ISBN: 978-81-903813-0-7

Blockchain

Foundations for Secure Intelligent Ecosystems



Edited & Contributed by

Members of Computer Science and Engineering and School of Computer Science

SWAMI VIVEKANANDA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTER SCIENCE

BLOCKCHAIN FOUNDATIONS FOR SECURE INTELLIGENT ECOSYSTEMS

2026



Swami Vivekananda University, Kolkata (Institutional Publisher)

Published by the Swami Vivekananda University (Institutional Publisher), Kolkata-700121, West Bengal, India.

The publishers reserve the right to prohibit any form of reproduction, distribution, or storage of this publication in a database or retrieval system, or in any other form or by any means, including mechanical, photocopying, recording, electronic, or otherwise. A computer system may be used to enter, save, and run the program listings (if any), but it is not permitted to duplicate them for publication.

Only the publisher may export this edition from India.

Swami Vivekananda University (Institutional Publisher), Kolkata-700121, India.

ISBN: 978-81-903813-0-7

First Edition: September, 2026

Publisher: Swami Vivekananda University (Institutional Publisher), Kolkata- 700121, India

Contact us: sourav@svu.ac.in

ACKNOWLEDGEMENT

We are writing to express our appreciation to Swami Vivekananda University in Kolkata, India, for all of their help and encouragement in producing this book, "**Blockchain Foundations for Secure Intelligent Ecosystems"** The university's dedication to supporting research and teaching has been important in determining the focus and substance of this publication. We really appreciate collaborative environment and resources of Swami Vivekananda University, Kolkata, which have made it possible for us to research and disseminate the newest developments in a variety of sectors. We hope that this book, which reflects our shared commitment to knowledge, advancement, and the pursuit of quality, will prove to be a useful tool for this prestigious institution as well as the larger academic community.

With sincere appreciation,

Dr. Ranjan Kumar Mondal (HOD), Dr. Payal Bose, & Sourav Saha
Assistant Professor
Swami Vivekananda University,
Kolkata, West Bengal, India

SWAMI VIVEKANANDA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTER SCIENCE

BLOCKCHAIN FOUNDATIONS FOR SECURE INTELLIGENT ECOSYSTEMS

Edited by

Dr. Ranjan Kumar Mondal

Dr. Payal Bose

Sourav Saha

and

Contributed by

Members of
School of Computer Science
Department of Computer Science & Engineering

Swami Vivekananda University
Telinipara, Barasat - Barrackpore Rd, Bara Kanthalia
West Bengal - 700121

Published Edition • 2026

Preface

In recent times, the importance of blockchain technology as a technological platform for building secure, transparent, decentralized, and intelligent digital ecosystems has been established. Blockchain, which was first developed as the basis for cryptocurrency systems, has gradually found its applications in many areas, ranging from finance and healthcare to logistics, smart cities, education, governance, and IoT. It is through the use of features like decentralization, immutability, cryptographically secure methods, consensus algorithms, and smart contracts that new mechanisms for building trust and secure information transfer have been possible. This book titled "**Blockchain Foundations for Secure Intelligent Ecosystems**" is purposely aimed at developing an understanding of the underlying fundamentals of the technology. Concepts, architectures, mechanisms, and applications associated with blockchain technology are carefully outlined in this book. Attention is paid to how blockchain can be integrated with other modern innovations such as AI, IoT, edge computing, and distributed intelligence. Also, the way through which blockchain technology contributes to improving data integrity, privacy, traceability, trust, and secure information exchange in intelligent ecosystems is explained. Furthermore, some of the most important challenges that include scalability, interoperability, energy efficiency, privacy, security, and regulation issues are critically analyzed. Emerging blockchain solutions are also covered in this book.

The book has been Intended as an academic and professional resource for students, researchers, educators, technology professionals, and practitioners interested in blockchain and its emerging applications. By combining foundational knowledge with contemporary technological perspectives and research directions, a comprehensive understanding of blockchain-enabled intelligent ecosystems is expected to be developed. It is hoped that this book will contribute to a deeper understanding of blockchain technology and its potential to support secure, trustworthy, and intelligent digital transformation.

Further comments and suggestions for improving the book will be gratefully received.

Dr. Ranjan Kumar Mondal (HOD), Dr. Payal Bose, & Sourav Saha
Assistant Professor
Swami Vivekananda University
Kolkata, West Bengal, India

List of Authors

(Members of School of Computer Science)

Swami Vivekananda University, Kolkata 700121, India

Prof. (Dr.) Somsubhra Gupta	Professor
Dr. Ranjan Kumar Mondal	Assistant Professor, (HOD)
Dr. Sanjay Nag	Associate Professor
Dr. Chayan Pal	Associate Professor
Dr. Abhijit Paul	Associate Professor
Dr. Pabitra Pal	Associate Professor
Dr. Payal Bose	Assistant Professor
Sourav Saha	Assistant Professor
Sumana Chakraborty	Assistant Professor
Sangita Bose	Assistant Professor
Lipika Mukherjee Pal	Assistant Professor
Bijaya Banerjee	Teaching Assistant
Pradip Sahoo	Teaching Assistant
Suparna Bandyopadhyay	Teaching Assistant
Nitai Das	Teaching Assistant
Ankita Saha	Teaching Assistant
Mitu Roy	Teaching Assistant
Soumyajit Jalua	Assistant Professor
Biplab Bhunia	Teaching Assistant
Rishov Chakraborty	Assistant Professor
Diwakar Pandit	Assistant Professor
Rohit Chawhan	Teaching Assistant
Mousumi Das	Assistant Professor
Subhadip Kowar	Teaching Assistant
Souvik Biswas	Teaching Assistant
Shreya Basu	Teaching Assistant
Dipanjan Das	Teaching Assistant
Manoj Kumar Rabidas	Teaching Assistant
Arunima Sinha	Assistant Professor
Amit Ghosh	Assistant Professor
Rinku Mahata	Assistant Professor
Kousik Dey	Assistant Professor
Somen Pandey	Teaching Assistant
Bishal Ray	Teaching Assistant
Tulika Paul	Teaching Assistant
Arafat Ali Mondal	Assistant Professor
Subhajit Mondal	Teaching Assistant

Saptashwa Karmakar	Teaching Assistant
Khakon Das	Assistant Professor
Subhadeep Guin	Teaching Assistant
Sagar Das	Teaching Assistant
Anirban Sinha	Assistant Professor
Ankita Mitra	Teaching Assistant
Krishnakash Giri	Assistant Professor
Sourasish Chakraborty	Teaching Assistant
Priya Majumder	Teaching Assistant
Disha Khan	Teaching Assistant
Arnab Das	Assistant Professor
Somenath Goswami	Assistant Professor
Sudeshna Saha	Teaching Assistant
Dibyani Samanta	Teaching Assistant
Sukanya Basu	Assistant Professor
Samrat Sahana	Teaching Assistant
Uddipta Modak	Assistant Professor
Swagata Das	Teaching Assistant
Jayanta Chowdury	Assistant Professor
Palash Baran Panja	Assistant Professor
Aditya Dutta	Assistant Professor
Tanmay Saha	Assistant Professor
Monalisa Bose Pal	Teaching Assistant
Sushovan Paul	Teaching Assistant

EDITORS' NOTE

This volume has been organized as a collaborative academic contribution of the School of Computer Science. The chapters are intended to provide a progressive understanding of blockchain foundations, security mechanisms, architectures, applications, and their integration with intelligent and emerging digital technologies. Contributors are encouraged to maintain consistent terminology, figure and table formatting, and chapter organization throughout the book.

About the Book

Blockchain Foundations for Secure Intelligent Ecosystems is a systematic and thorough examination of blockchain technology and its changing position in the context of secure, decentralized, transparent, and intelligent digital ecosystems. Starting from the position of blockchain being the underlying technology of cryptocurrency systems, blockchain technology has matured to become a flexible distributed technology with vast possibilities in different fields such as finance, health care, supply chain management, smart cities, governance, education, and the Internet of Things (IoT). Some of the key technologies that are covered in this book include fundamental concepts of Distributed Ledger Technology (DLT), cryptography, consensus, blockchain architecture, smart contracts, decentralized applications, security, privacy, and data integrity. These technological foundations are analyzed in the context of the needs and challenges of the modern digital world, where the issues of trust, transparency, accountability, secure data transfer, and decentralized decision-making have become more relevant than ever before. In addition, different types of blockchains and their applicability to the development of secure and scalable digital infrastructures are covered. Of particular interest in this book is the role that blockchain plays in the combination with other emerging technologies like Artificial Intelligence (AI), the Internet of Things (IoT), edge computing, and distributed intelligence. Their combination is considered as one of the ways to create intelligent ecosystems with coordinated data, computational power, digital services, and autonomous processes.

Apart from discussing the technical aspects and applications of blockchain technology, this book also discusses the important challenges associated with blockchain implementation. Scalability, interoperability, energy consumption, cybersecurity, privacy, data governance, regulatory issues, and decentralized system management are some of the problems that have been identified. Important research directions as well as novel blockchain applications have also been highlighted in order to offer a wider perspective of how blockchain technology can be used in developing intelligent digital ecosystems in the future. This book is aimed at addressing the needs of undergraduate and postgraduate students, researchers, academics, professionals, and practitioners who wish to acquire conceptual and practical knowledge of blockchain technology and its new applications. By covering the foundational concepts, recent technological developments, practical insights, and research directions, this book aims to build a holistic perspective on how blockchain can be leveraged to develop intelligent and secure digital ecosystems.

TABLE OF CONTENTS

Part – I: Foundational Paradigms

Chapter 1: Genesis of Blockchain and Decentralized Intelligence 1
Chapter 2: Distributed Ledger Architectures and Ecosystem Design 20
Chapter 3: Cryptographic Principles for Secure Computation 34
Chapter 4: Consensus Mechanisms and Algorithmic Trust 51
Chapter 5: Artificial Intelligence Foundations: Models and Learning Paradigms 66
Chapter 6: Conceptual Frameworks for Convergence of Blockchain and AI 82

Part – II: Architectures, Governance, and Security

Chapter 7: Architectures of Decentralized Artificial Intelligence 95
Chapter 8: Federated and Distributed Learning Systems 113
Chapter 9: Blockchain-Enabled Federated Intelligence 130
Chapter 10: Smart Contracts and Autonomous Decision Frameworks 150
Chapter 11: Decentralized Autonomous Organizations and Governance Models 164
Chapter 12: Tokenized Incentive Mechanisms for Intelligent Networks 177
Chapter 13: Security Challenges in Blockchain and AI Systems 192
Chapter 14: Privacy-Preserving Mechanisms in Decentralized Intelligence 206
Chapter 15: Trustworthy AI: Fairness, Transparency, and Accountability 221

Part III: Integration, Applications, and Future Frontiers

Chapter 16: Blockchain-AI Integration Frameworks and Marketplaces 237
Chapter 17: Applications Across Healthcare, Finance, IoT, and Cybersecurity 256
Chapter 18: Future Directions: Decentralized AGI, Quantum-Resistant Systems, and Ethical Horizons 276

References

PART – I

FOUNDATIONAL PARADIGMS

CHAPTER 1

GENESIS OF BLOCKCHAIN AND DECENTRALIZED INTELLIGENCE

Contributed by: Prof. (Dr.) Somsubhra Gupta and Nitai Das

1.1. INTRODUCTION

Blockchain is an ever-growing chain of linked records (blocks) secured through the use of cryptography. Blockchain was invented in 2009 by Satoshi Nakamoto. The decentralized model of trust enabled by the blockchain integrates many different elements to produce a secure and trustworthy environment for untrusted users to interact amongst themselves. Unlike centralized models or traditional computer systems where a central authority is required to verify transactions and protect data integrity. Adding to its effect on digital banking, the transition now provides possibilities for new approaches to Supply Chain Management, Healthcare, Smart Cities, Digital Identity, Governance, Cyber Security and the rapidly growing field of Artificial Intelligence. The development of blockchain technology is not however, a unique independent occurrence; rather it is ultimately the result of interrelated, interdisciplinary research within the fields of cryptography, computer networking, distributed systems, game theory, economics and the field of information security. Each of the aforementioned areas of research have contributed to develop technologies and ideas to solve what has traditionally been one of computer science's greatest challenges: Providing a method for users on the Internet to establish trust and coordinate actions without a central entity that exerts some level of authority. Through the emergence of various innovations, including Merkle Trees, Byzantine Fault Tolerance, Peer-to-Peer Networking, and Distributed Consensus Algorithms, the various streams of research developed over the years have finally merged together successfully into a true decentralised system that was capable of providing secure, consensus-based transactions on an open network while preventing double spending. In 2008, the release of Bitcoin was the first realisation of such a system. Following the introduction of Bitcoin, Blockchain technology has developed into more than just being a digital currency provider. Developments such as the creation of smart contracts, DApps, DeFi, tokenized digital assets, DAOs, and multi and 50 chain ecosystems provide the basis for how blockchain is the foundational technology of our new Decentralized Digital Economy. Recent advances in AI are also producing a new wave of innovation whereby Blockchain acts as a trusted value generator. With this new convergence between AI and blockchain technology, we can witness an emerging trend of decentralized intelligence. This trend is characterized by the combination of AI's ability to learn, evaluate, and make decisions with the transparency, reliability, verifiability, and security of blockchain so as to create independent, resilient, and trustworthy digital

environments. To assist the rest of the body of work, Chapter 1 provides a complete foundation for the discussion, focusing specifically on the historical development, the theoretical basis, the technological developments, and the emerging nexus of blockchain and decentralized intelligence. It establishes a conceptual framework in order to better understand advanced architectures, decentralized AI architectures, real-world applications, governance, security concerns, and the future development of intelligent decentralized systems as discussed in the chapters that follow.

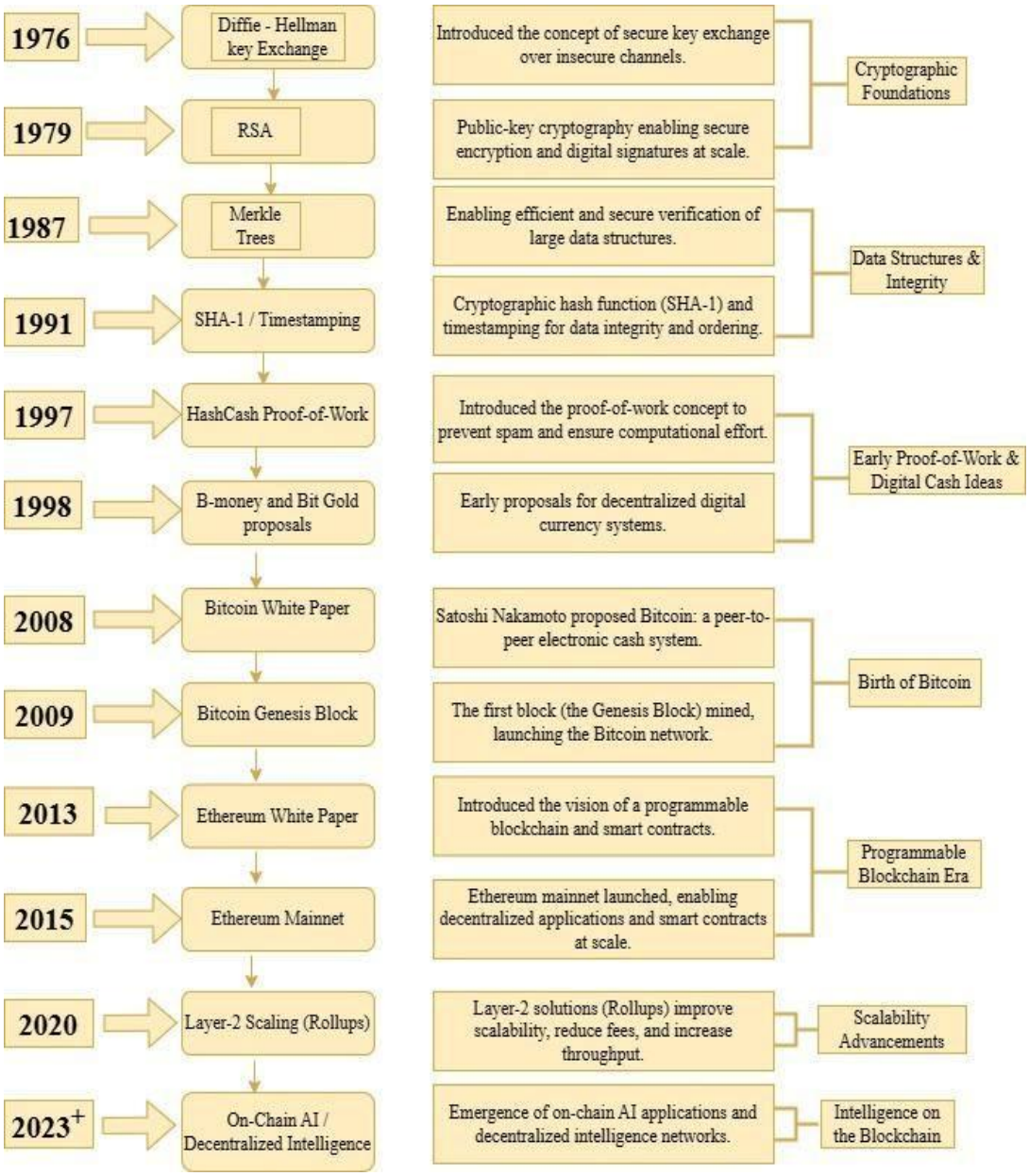


Figure 1: Timeline of Key Milestones Toward Decentralized Intelligence

1.2 HISTORICAL GENESIS OF BLOCKCHAIN

In 1976, a paper was produced on “New Directions in Cryptography” introducing the concept of distributed ledger. As cryptography advanced, Stuart Haber and Scott Stornetta's paper "Hot to Time-Stamp a Digital Document" introduced the idea of timestamping the data rather than

the medium. Another significant idea known as "Electronic cash" or "Digital Currency," which originated from a model put forth by David Chaum, also aided in the development of the Blockchain concept, which was succeeded by protocols like e-cash schemes that introduced double spending detection. In 1997, Adam Back developed another notion called “hashcash” which suggested a means to control spam emails. This resulted in Wei Dai's idea to create "b-money," which is based on a peer-to-peer network. In 2008, Nakamoto published a solution to the problem of ‘double spending,’ noting that a digital representation of money should not exist more than once, or it should not be copied or spent more than once. The paper introduces a public ledger concept to solve this problem, i.e., by allowing electronic coins to have their transaction history verified through the ledger as an added measure to verify that they had not yet been used before. A few months later, Satoshi Nakamoto released the first implementation of the Bitcoin system (an open-source program). In January 2009, Nakamoto mined the first Bitcoins and created the very first Bitcoin blockchain network. Although the identity of the inventor of Bitcoins is still unknown, Bitcoins are still mined, marketed, and supported by a large global community. Many of which contribute to discovering and fixing many source code errors and bugs.

Although many alternatives to bitcoin exist (including dogecoin and litecoin), the majority of the cryptocurrency market is dominated by bitcoin. While it continues to be the most popular cryptocurrency available, its popularity is largely due to the reason that users feel vested in bitcoin's overall unity as a payment method through its relative transparency. Thus far, bitcoin has grown significantly; in early 2013, many investors were beginning to invest in bitcoin-themed companies (or bitcoin technology start-ups). Users can convert their bitcoins into fiat money for use at a brick-and-mortar retailer, e-commerce site, or by simply exchanging them for goods and services. Additionally, users may electronically send their bitcoins to other users via their computer, smartphone, or third-party online application utilizing software which functions as a wallet.The Ethereum platform was introduced in 2015, allowing blockchain to function with contacts and loans. It was predicated on a smart contract mechanism that guaranteed the two parties would carry out an action. Ethereum gained widespread popularity as a result of its capacity to provide a quicker, safer, and more effective environment.

1.2.1 Foundation of Modern Cryptography

In 2015, the Ethereum platform was released, enabling blockchain to work with loans and contacts. It was based on a smart contract system that ensured both parties would do a certain action. Ethereum's ability to offer a faster, safer, and more efficient environment led to its worldwide adoption. Whitfield Diffie and Martin Hellman's 1976 article, which showed that two people may establish a shared secret via a public, unsecured channel without ever exchanging a private key beforehand, is where the quest for digital trust began. The realization that mathematical trapdoor functions may replace physical secrecy drastically changed the course of cryptography. The first workable public-key cryptosystem and the notion of the digital signature a mathematical demonstration that a message was written by the owner of a particular private key were made possible within two years when Ronald Rivest, Adi Shamir, and Leonard Adleman operationalized the idea in the RSA algorithm.

Cryptographic Hash Functions

One of the most basic primitives in the blockchain toolset is the cryptographic hash function. A hash function satisfies three essential security requirements by mapping an arbitrary-length input to a fixed-length output (the digest or hash). First, it must be collision-resistant: it must be computationally infeasible to identify two distinct inputs that produce the same hash. Second, it needs to be resistant to pre-images, meaning that any input that would generate a hash must be impossible to retrieve. Third, it must display the avalanche effect: a single-bit change in the input must, on average, flip half the bits of the output, rendering the digest unpredictable and pseudo-random. These characteristics were gradually strengthened by the U.S. National Security Agency's Secure Hash Algorithm family, which included SHA-1 (1993), SHA-256 (2001), and the later SHA-3 Keccak standard. SHA-256 is notably used twice in succession (SHA-256d) in Bitcoin's proof-of-work method to generate the block hashes that characterize the chain. In contrast, Ethereum chose to use Keccak-256 for its internal hashing, which reflects various performance trade-offs on the Ethereum Virtual Machine (EVM).

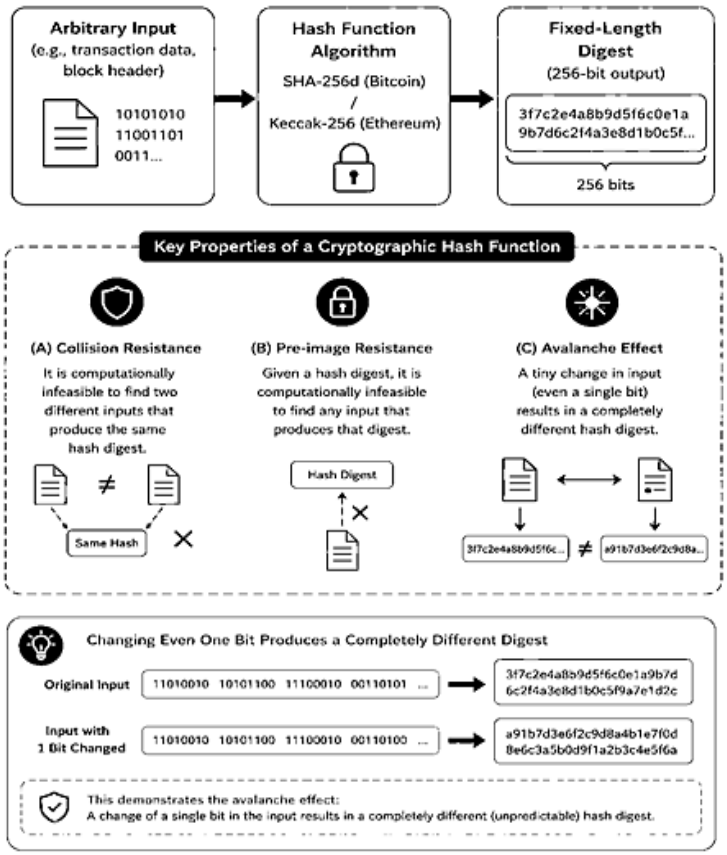


Figure 2: Anatomy of a Cryptographic Hash Function

Public-Key Infrastructure and Digital Signatures

A digital signature system involves three algorithms: key generation (creating a public/private key pair), signing (using the private key to form a signature over a message), and verification (using the public key to check the signature's authenticity). Digital signatures are the authoritative proof of ownership in blockchain environments. In order to initiate a smart contract in Ethereum or spend a UTXO in Bitcoin, the starting party must provide a legitimate signature under the private key associated with the asset's address. Without possession of the private key, creating a valid signature is computationally indistinguishable from impossible. Because of its better performance features and smaller key sizes than RSA, Bitcoin uses the

Elliptic Curve Digital Signature Algorithm (ECDSA) over the secp256k1 curve. Secp256k1 is also used by Ethereum. In anticipation of the potential danger to elliptic-curve cryptography posed by large-scale quantum computers, more contemporary systems, such as Solana and post-quantum contenders, have investigated EdDSA over Curve25519 and lattice-based signature methods, respectively.

Table 1: Core Cryptographic Primitives and Their Blockchain Roles

Primitive	Core Property	Blockchain Role	Primary Algorithm(s)
Hash Function	Collision & Pre-image Resistance	Block linking, Merkle trees, PoW puzzle	SHA-256d, Keccak-256
Merkle Tree	Logarithmic Integrity Proofs	Transaction commitment in block header	Recursive SHA-256d / Keccak
Digital Signature	Non-repudiation, Authentication	Transaction authorization, identity	ECDSA (secp256k1), EdDSA
Public-Key Crypto	Asymmetric Key Exchange	Address generation, key derivation	ECDH, RSA (legacy)
Zero-Knowledge Proof	Verifiable Computation w/o disclosure	Layer-2 validity, privacy	zk-SNARKs (Groth16), STARK

1.2.2 Distributed Systems and the Problem of Consensus

Cryptographic primitives determine the characteristics of individual data items, providing answers to queries like who signed a message or if a file has been altered. They do not, however, address the more difficult problem that blockchain has to solve: how can a network of geographically scattered, mutually suspicious nodes come to an agreement on a single, canonical ordering of events? This is the consensus problem, and its formal treatment constitutes one of the most intellectually rich fields of computer science.

Distributed Database

With its decentralized and safe approach to data management, distributed systems and blockchain technology are revolutionizing industries. This article examines how they work together, emphasizing how blockchain improves distributed system architectures' transparency and trustworthiness.

The Byzantine Generals Problem

The Byzantine Generals Problem was first formulated by Leslie Lamport, Robert Shostak, and Marshall Pease in 1982. A collection of Byzantine army generals, each in charge of a division, are depicted in the allegory as having to decide on a shared strategy. It's possible that some of the generals are traitors who deliberately try to create confusion by providing conflicting messages. Regardless of the traitors' behavior, the problem asks what minimal percentage of

loyal generals is needed to ensure that the loyal generals come to an agreement. The mathematical result is striking: if more than one-third of participants are Byzantine (malicious or defective), no deterministic synchronous algorithm can ensure consensus. This result set the theoretical ceiling on Byzantine Fault Tolerance (BFT) for traditional distributed systems. Conventional BFT protocols, such as Miguel Castro and Barbara Liskov's 1999 publication Practical Byzantine Fault Tolerance (PBFT), reach consensus under this one-third restriction, but at the expense of $O(n^2)$ message complexity, making them unsuitable for large, open networks.

Peer to Peer Architecture

In a peer-to-peer (P2P) network, each computer acts as both a client and server; allowing for distribution of data, sharing of files, and distributing processing power. There are numerous characteristics, advantages, types, components, bootstrapping procedures, methods to organize data, routing algorithms, problems, security features, and applications for the P2P network topology that will be discussed within this paper. Characteristics of peer-to-peer (P2P) networks:

1. Decentralization; P2P networks allow for nodes to communicate and share resources without a central authority.
2. Scalability; P2P networks can easily scale by adding numerous nodes without the need for a centralized infrastructure.
3. Fault-tolerance; P2P networks are fault-tolerant, since there is no central server, the P2P network can continue to function with missing nodes.
4. Resource sharing; P2P network users have the ability to directly share files, data, and computer resources among peers.
5. Autonomy; Each node in a P2P network has independent control of its own resources and decision-making capabilities for their own autonomy, increasing the overall durability and flexibility of a P2P network.

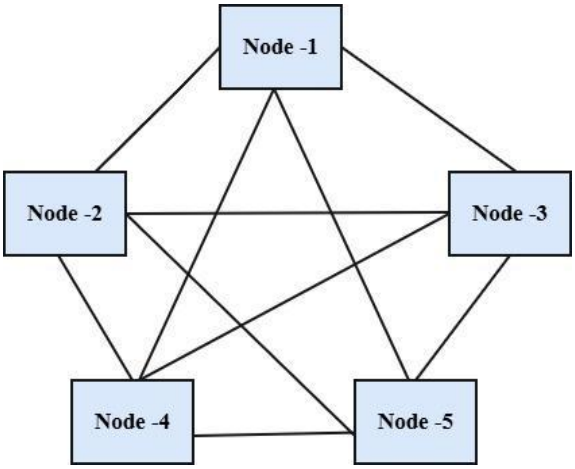


Figure 3: P2P architecture

1.2.3 Digital Currency before Blockchain

eCash is referred to as "Electronic Cash," a digital currency method that allows transactions to be completed online from any location. It is a simpler method of payment and is built on the

Peer-to-Peer network's blockchain technology (Digital Signatures). Every transaction and exchange is kept in a particular digital database. It is the alternate payment system to pay for bills, products, and services without the usage of paper or coin currencies. Digital currency, debit cards, electronic cases, electronic checks, and credit cards are examples of applications for electronic or digital cash.

Features of Electronic Cash:

- Decentralized: This indicates that distributed ledgers are used by a centralized entity to govern them. Because the consumer might not need to trust anyone, it lessens trust difficulties. It enhances consistency and performance. It makes transactions irreversible.
- Transparency: This implies that nothing will be concealed from the participants and that every transaction is visible and understandable. This strengthens the trust and faith in electronic cash.

Illustrations of eCash

Famous cryptocurrencies Bitcoin and Ethereum are founded on the principles of cryptography and blockchain, making them decentralized, and safe, and removing the third party from involvement with transactions. They are not authorized/backed by anybody properly or legally.

Cryptocurrencies: Known as digital currency, they are created utilizing encryption technology and are highly decentralized due to their characteristics. The primary benefit of cryptocurrencies is their transparency and speed of money transfers. However, mining is more expensive, and cryptocurrency movements are highly erratic.

Central Bank Digital Currencies: Known as fiat currency, it is completely approved by the legal entity. Unlike other digital currencies, it does not conceal payment transactions. It is directly correlated with the financial economy of the nation.

Stablecoins: Stablecoins are the coins that are entirely stable according to the currency rate of a given country, such as Tether, TrueUSD, Binance USD, Dai, etc. Its exchange rate is fixed to the fiat currency of a nation such as the United States of America. Due to its extreme safety, it is most in demand during periods of strong investment.

1.2.4 Bitcoin and the Birth of Blockchain

By substituting computational expense for identity-based voting, Bitcoin's consensus mechanism now known as Nakamoto Consensus avoids the traditional BFT limitation. Bitcoin requires each candidate block to be accompanied by evidence that the miner put in a significant amount of computing effort to locate it, as opposed to counting votes from a known set of participants. In particular, the miner must determine a nonce value such that the block header's SHA-256d hash is less than a dynamically modified target value T. The only method to find such a nonce is through brute-force iteration, which uses actual energy because hash functions are pseudo-random. To find a valid nonce, the estimated number of hash calculations is inversely proportional to T. The Bitcoin protocol makes sure that the cost of creating a valid block scale with the network's overall hash rate by modifying T every 2,016 blocks, or roughly two weeks, to maintain a ten-minute average block time. Therefore, the well-known 51% attack threshold requires an attacker who wants to change history to out-compute the whole honest mining

network. The economic cost of such an attack greatly outweighs any conceivable return in a suitably big, geographically dispersed mining pool.

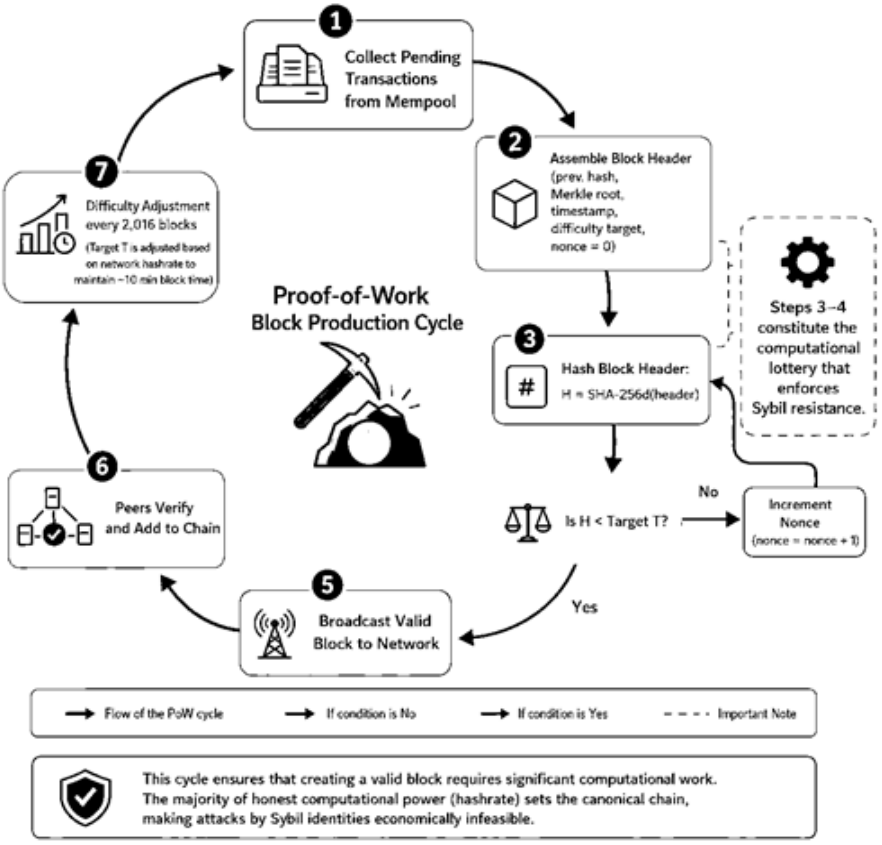


Figure 4: Nakamoto Consensus: The Proof-of-Work Block Production Cycle

The Genesis Block and Network Bootstrap

There is only one hard-coded starting point for the Bitcoin blockchain: On January 3, 2009, Satoshi Nakamoto mined Block 0, also referred to as the Genesis Block. The text string "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" is included in the coin-base transaction, which is how miners collect their block reward. The British newspaper's headline The Times has two functions: it encodes an implicit commentary on the vulnerability of centralized financial institutions, which is embedded in the chain's first transaction and has since become one of the most quoted passages in computer science history, and it provides an unforgeable timestamp proving the block was mined no earlier than that date. The prev hash of the Genesis Block is set to all zeros, a unique sentinel value that denotes the lack of a predecessor. Each succeeding block in the chain may be traced back to this anchor in an uninterrupted cryptographic lineage. As the white paper gained popularity, the network grew naturally from a small number of nodes run by cryptography enthusiasts on mailing lists. By 2010, a working market had developed, and the first real-world transaction, the purchase of two pizzas for 10,000 BTC, showed that the system could operate as a legitimate currency.

1.2.5 Evolution of Blockchain

From Bitcoin to a Programmable World Computer: The Evolution of Blockchain

Vitalik Buterin was nineteen years old when he circulated the Ethereum white paper in late 2013. His main realization was that Bitcoin's blockchain could be expanded: a complete virtual machine could be incorporated into a blockchain, enabling users to implement arbitrary programs and smart contracts whose execution was ensured by the network's consensus mechanism, rather than a scripting language created specifically for financial transactions. The solution was designed as a “world computer”: a worldwide, censorship-resistant, trustless compute layer on which any application might operate.

1.3 FOUNDATIONS AND ARCHITECTURE OF BLOCKCHAIN

Blockchain technology works on the concept of decentralized databases where these databases exist in multiple computers and every copy of these databases are identical. Businesses store their data in centralized systems, which makes them easy targets for hackers. The decentralized characteristic of blockchain’s structure allows it to be a resilient form of technology. Considerably, a blockchain can be perceived as a peer-to-peer service over the internet. There are three main layers that make up the architecture of blockchain; these are applications, decentralized ledger and peer-to-peer (P2P) network respectively from top to bottom respectively. The uppermost layer (Applications) holds the application software; the P2P Network holds the industry-standard protocol that creates the various applications; while the Decentralized Ledger holds the actual transactions executed on the blockchain. For example, the bitcoin wallet application software creates and stores the user's private and public key enabling them to control their unspent bitcoins. Users can also see their transactions through a user-friendly interface that is provided by the application layer.

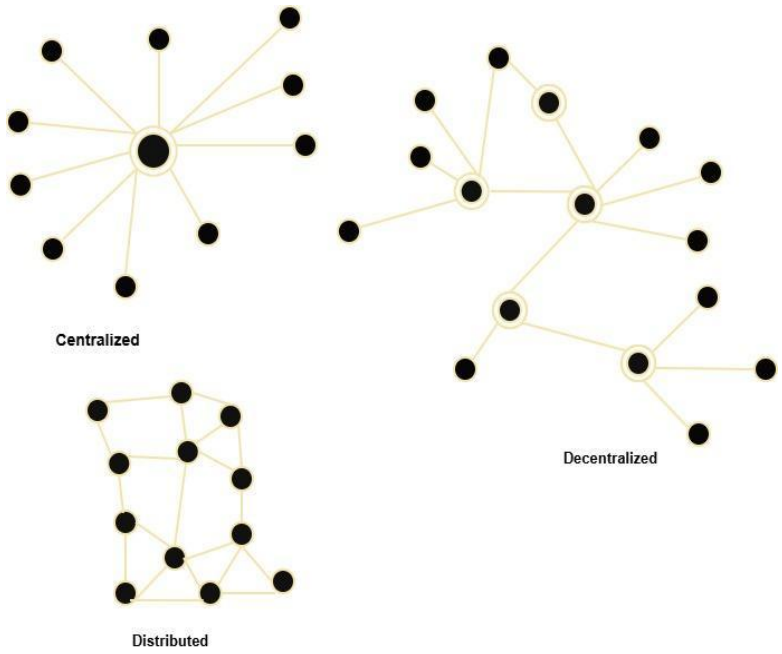


Figure 5: Centralized, Decentralized and Distributed

Decentralized Ledger is the intermediary layer of a blockchain architecture that confirms a consistent and temper-proof global ledger. This layer allows transactions to be organized into blocks that are connected to each other cryptographically. Token exchanges between two

parties are referred to as transactions, and each transaction must pass a validation process in order to be deemed genuine. The process of assembling transactions into a block that is appended to the end of the existing blockchain is known as mining. Blockchain employs a proof-of-work method to identify which chain has taken the longest to construct and to guarantee consensus among all nodes in order to establish the legitimacy of the blockchain. The Peer-to-Peer Network, the lowest layer of the blockchain architecture, is where different types of nodes play different functions and exchange different messages to keep the Decentralized Ledger up to date. It provides application interfaces on top of the blockchain and is utilized for keeping the cryptocurrency secure. This program can be installed on your PC or mobile devices or alternatively can be hosted on a third-party platform.

1.3.1 Distributed Ledger Technology (DLT)

Public Ledger: Shared Ledger

1. Records all transactions across the business network.
2. Shared between participants.
3. Participants have their own copy through replication.
4. Permissioned, so participants see only appropriate transactions.
5. It is the shared system of record.

Blockchain as public ledger

1. Distributed ledgers are held, restructured, and managed by individuals called nodes.
2. Every node builds the database on its own.
3. Every transaction that takes place on the network is processed, and each node generates a conclusion regarding the database's development.
4. Based on the transaction, voting is carried out on the changes completed on the database. All nodes participate in the voting, and if at least 51% of them agree, the new transaction is accepted in the database.
5. The nodes then update the database versions so that every device or node has the same version.

Benefits of Distributed Ledgers

1. Highly transparent, secure, tamper-proof, and unchangeable. Records cannot be changed by anybody else once they are entered into distributed ledgers.
2. A third party is no longer required.
3. Inherently decentralized
4. Extremely clear

1.3.2 Consensus Mechanisms

Proof of Work (PoW)

Cynthia Dwork and Moni Naor initially proposed the concept of Proof of Work (PoW) in 1993, and Satoshi Nakamoto used it in the Bitcoin paper in 2008. For most of the cryptocurrencies now in use, Proof of Work consensus is the preferred method. In a 1999 article, Markus Jakobsson and Ari Juels coined the phrase "proof of work. "Principle: A solution that is hard to locate but simple to confirm. In an environment where nodes don't trust one another, a

consensus mechanism's goal is to get all of the nodes to agree, or trust one another. The new block is then added to the blockchain after every transaction in it has been verified. Note that, the block will get added to the chain which has the longest block height (see blockchain forks to understand how several chains can exist at a moment of time). In order to add a block to the network, miners specialized computers on the network perform computational labor to solve a challenging mathematical problem; this process is known as Proof-of-labor. With time, the mathematical issue becomes more difficult.

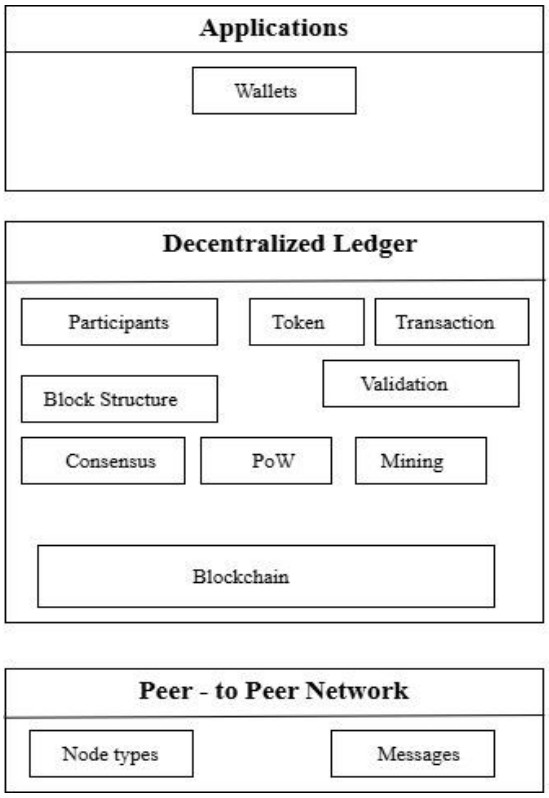


Figure 6: Foundations of Blockchain Architecture

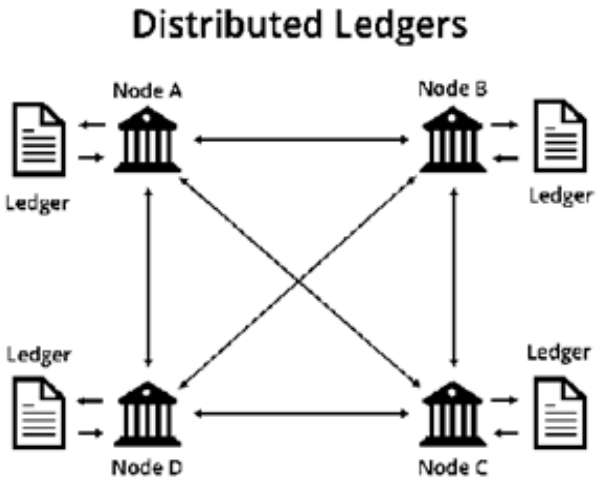


Figure 7: Illustration of distributed ledger

Working of POW (Proof of Work)

The Proof of Work consensus mechanism creates new blocks in the Bitcoin blockchain by resolving a computationally difficult challenge. The process is colloquially referred to as "mining," and the network nodes that participate in it are referred to as "miners." Economic payoffs, in which rival miners are rewarded with 12.5 bitcoins (as of the writing of this article; this payout will eventually be cut in half) and a modest transaction fee, serve as the incentive for mining transactions. The process of confirming the transactions in the block to be added, structuring these transactions in a chronological order in the block and notifying the freshly mined block to the entire network does not take much energy and time. The energy consuming aspect is solving the 'hard mathematical problem' to link the new block to the last block in the valid blockchain. When a miner ultimately finds the proper solution, the node broadcasts it to the complete network at the same time, getting a cryptocurrency prize (the reward) offered by the PoW protocol. At the time of writing this article, mining a block in the bitcoin network awards the winning miner 12.5 bitcoins. The quantity of bitcoins won halves every four years or so (that's how the bitcoin network is constructed). Thus, given the current rate and growth, the next reduction in the quantity of bitcoin is expected to occur around 2020–2021.

The fact that Block GFG1 is related to Block GFG2 through its hash number is essential. The significance comes in the fact that this 'hash number' connects a new block to the last block in the valid blockchain. If, on the other hand, the Block GFG1 Hash number on Block GFG2 had a different hash number than Block GFG1 they would not line up, and Block GFG2 would not be validated. The blockchain's initial block, known as the Genesis Block, lacks a Prev Block Hash value. It is nearly impossible to change a block (which can only be done by creating a new block with the same predecessor) because doing so necessitates regenerating all successors and performing the work they contain, which amounts to solving the full chain of "hard mathematical problems." By doing this, the blockchain is shielded against manipulation.

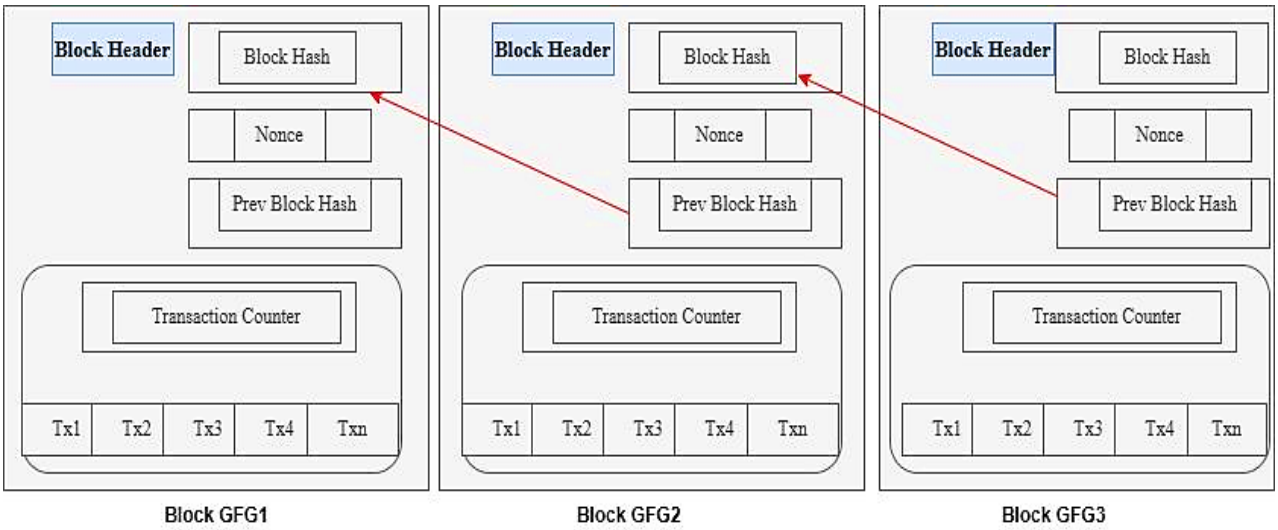


Figure 8: Proof of Work

Proof of Stake (PoS)

One kind of algorithm called Proof of Stake (PoS) seeks to accomplish distributed consensus in a blockchain. This method for attaining consensus was created first by Quantum Mechanic and

written about by Sunny King and his team in a paper that followed this initial proposal. The final product, Peercoin is a currency that has been designed using Proof-of-Stake (PoS) as its consensus mechanism. In this case, the stake is an amount of money that we are putting on something that would eventually come true.By doing this process, you have become a "staker," and as we move through the article, we will more specifically define the term "stake."

1.3.3 Smart Contracts and Decentralized Applications

As one of the initial contributors to the Bitcoin software, Vitalik Buterin is part of an increasing number of engineers who believe that Bitcoin has not yet reached the full potential of blockchain technology in a future where innovation is the norm. Concerned about Bitcoin's limits, Buterin started working on what he felt would be a customizable blockchain that can do many purposes in addition to being a peer-to-peer network. In 2013, Ethereum emerged as a new public blockchain with more features than Bitcoin. This development has proven to be a turning point in the history of blockchain technology. Buterin set Ethereum apart from the Bitcoin Blockchain by introducing a feature that lets users store other assets like contracts and slogans. The new feature increased Ethereum functionalities from being a money to becoming a platform for developing decentralized applications as well. Officially released in 2015, Ethereum blockchain has evolved to become one of the greatest implementations of blockchain technology given its ability to allow smart contracts used to execute numerous purposes. Ethereum blockchain technology has also succeeded in gathering an active developer community that has seen it construct a true ecosystem. Because the Ethereum blockchain can handle decentralized apps and smart contracts, it handles the most transactions per day. In the cryptocurrency world, its market capitalization has also grown considerably.

1.3.4 Scalability and Interoperability Challenges Include

Blockchain technology, which promises decentralized trust without reliance on a central authority, has emerged as a disruptive force in a variety of industries, including supply chain management, healthcare, and banking. However, as use increases, scalability and interoperability, two enduring and closely related challenges, limit blockchain's potential to become a widely used, enterprise-grade tool. This chapter addresses the nature of these difficulties, their underlying causes, and the technical techniques being created to overcome them.

1.4. GENESIS OF DECENTRALIZED INTELLIGENCE

1.4.1 Defining Decentralized Intelligence

The architecture and administration of intelligent systems are changing due to the convergence of blockchain technology, smart contracts, and decentralized artificial intelligence (DAI). It is becoming more and more important to address privacy in these technologies' architecture, usage models, and related dangers as they develop quickly. This comprehensive analysis of the literature looks into DAI systems' architectural trends, governance structures, practical uses, and enduring problems. It lists popular designs such as decentralized verification techniques, smart contract-based incentive systems, and federated learning combined with consensus protocols. The review, which draws on a wide range of current literature, highlights

implementations in a number of industries, including healthcare, banking, IoT, autonomous systems, and intelligent infrastructure, all of which have made substantial contributions to security, privacy, and cooperative innovation. Despite these developments, DAI systems face persistent problems including scale constraints, privacy trade-offs, and difficulty with regulatory compliance.

1.4.2 Evolution of Artificial Intelligence

Contemporary AI systems are virtually entirely centralized. A small number of IT companies with access to enormous computational resources and private data train and run large language models, recommendation engines, autonomous trading systems, and medical diagnostic tools. This centralization causes numerous forms of systemic risk.

1. Opacity: Major AI systems have opaque inference architecture, model weights, and training data. Users are unable to independently confirm whether a model was trained on the purported data, has not been adjusted to display particular biases, or is producing honest results rather than ones that are optimized for profit or engagement.
2. Accountability Gaps: Credit scoring, medical diagnosis, legal decisions, and political discourse are all increasingly impacted by AI-generated material, decisions, and recommendations; nevertheless, the organizations in charge of these systems have few accountability mechanisms.
3. Single Points of Failure and Control: Centralized AI providers constitute single points of failure for the services that depend on them, and single points of power that regulators, adversaries, or unscrupulous personnel might potentially use to alter outputs.
4. Resource Monopoliza: Researchers and organizations working in resource-constrained environments face greater obstacles to entrance as AI training has turned into a winner-take-all race for data and computation, solidifying established advantages.

1.4.3 Multi - Agent Systems

A multi-agent system (MAS) is a computer framework made of several autonomous, interacting software agents, each capable of seeing its surroundings, making independent decisions, and acting to achieve individual or collective goals. Blockchain technology acquires intelligent, adaptive agents that can automate complicated decision-making without central oversight, while MAS gains a decentralized, tamper-resistant infrastructure for coordination, trust, and transaction settlement. Traditional MAS solutions frequently depend on a centralized coordinator or trusted third party to oversee agent interactions, validate transactions, and handle disputes, a design that introduces single points of failure and limits scalability. Blockchain solves these shortcomings by providing:

1. Decentralized Trust: Since the blockchain itself verifies and logs interactions, agents can transact and coordinate without depending on a central authority.
2. Transparency and Auditability: Every transaction or agent action is documented on an unchangeable ledger, enabling post-event behavior verification for all parties.
3. Secure Value Exchange: Smart contracts allow agents to exchange digital assets, data, or services on their own, eliminating the need for manual settlement.
4. Fault Tolerance: the distributed nature of blockchain decreases the danger that a single compromised or failed node disrupts the entire system.

1.4.4 Edge Intelligence

Instead of depending exclusively on centralized cloud infrastructure, edge intelligence refers to the implementation of artificial intelligence and data-processing capabilities directly on edge devices, sensors, gateways, IoT nodes, or local servers. Edge intelligence allows autonomous agents to see, think, and act locally at the network's edge when paired with multi-agent systems (MAS) and blockchain, while blockchain guarantees that their choices and interactions stay safe, verifiable, and synchronized throughout the larger system. Multi-agent systems working in dispersed environments such as smart grids, autonomous car networks, or industrial IoT create huge volumes of data at the network edge. Latency, bandwidth strain, and privacy concerns arise when all of this data is sent to a central cloud for processing. Edge intelligence addresses these constraints by allowing agents to:

1. Process Data Locally: Lowering latency and bandwidth consumption by eliminating the need to send raw data to remote servers.
2. Make Real-Time Decisions vital for time-sensitive applications, such as industrial automation or driverless cars, where delays could have an impact on operations or safety.
3. Preserve Privacy: Only aggregated findings or conclusions can be shared with the larger network when sensitive data is examined on-device.
4. Operate Under Intermittent Connectivity: Even in the event of unreliable connections to central infrastructure, agents can still operate.

Role of Blockchain in Edge-Intelligent MAS

By offering a decentralized trust layer for coordinating dispersed, locally intelligent agents, blockchain enhances edge intelligence:

1. Verifiable Agent Decisions: Edge agents' actions can be recorded on-chain, resulting in an unchangeable audit trail that promotes accountability without the need for centralized supervision.
2. Secure Model and Data Sharing: Through blockchain-mediated transactions, agents can exchange trained models, insights, or data summaries with one another, guaranteeing provenance and integrity.
3. Incentivized Collaboration: By rewarding edge agents for providing data, computational resources, or model updates, smart contracts can promote cooperative behavior throughout the network.
4. Decentralized Consensus on Local Decisions: Blockchain-based consensus algorithms can validate results without a central controller when several edge agents need to agree on a common action (like load balancing in a smart grid).

1.4.5 Blockchain as an AI Trust Layer

Blockchain offers a convincing set of tools to handle some of these issues, but it cannot fix all of them. In order to create an unchangeable audit trail that enables independent verification of model provenance, on-chain model registries can store cryptographic commitments (hashes) of model weights and training datasets. In order to enable federated learning and data-marketplace patterns that enable AI models to be trained on distributed private datasets without centralizing that data, smart contracts can enforce data-sharing agreements between

parties that do not trust one another. Decentralized Oracle Networks, of which Chain-link is the most extensively used example, offer a tamper-resistant way to transfer off-chain data (such as real-world prices, weather readings, sports results, and AI inference outputs) into the deterministic on-chain environment. Any application that needs a smart contract to react to actual events including AI-driven smart contracts that need to act on model inference results requires this infrastructure.

1.4.6 Decentralized Intelligence

A shared, replicated database that is synchronized across network participants is called a decentralized ledger. It keeps track of the transactions that take place between network users. The ledger is responsible for keeping records of transactions among the participants. With the exception of storing data in the form of tokens or cryptocurrencies, blockchain technology shares characteristics with databases. The initial stage in entering transactions into the ledger is to organize the recently validated transactions into blocks. Any participant in the blockchain can gather fresh transactions and construct blocks that can be appended to the blockchain. A block largely contains transactions and has references, timestamps and the nonce. Depending on their position inside the blockchain network, nodes carry out a variety of tasks. A node can be considered a miner when it proposes and validates transactions and undertakes mining to provide consensus to protect the blockchain. It can perform operations such as simple payment verification etc., and functionalities based on the blockchain utilized. A consensus algorithm that confirms the accuracy of data is known as proof of work. Hashcash, for instance, serves as a proof of work for Bitcoin transactions. In order for the network to accept the transactions in the block, miners must finish a proof of work. The blockchain network's security and consensus are guaranteed by proof of work. A block is given a hash (id) throughout the verification process. This hash is appended to the current block of transactions in order to verify the subsequent block. Add a nonce to a random integer that can only be used once to the end of the subsequent block in the subsequent step. Hash function is used to change this random number to generate a string that contains a number of zeros in front of it. Because proof of work always depends on the miners' incentives, it is expensive to maintain and may have security and scalability problems in the future. A sophisticated mechanism known as "proof-of-stake," which is profitable to implement, determines who gets to update the consensus and prevents undesired blockchain forking. No confidential information is exchanged in a blockchain network and all the transactions are available to every node in the network. This peer-to-peer network can be constructed on any physical infrastructure and doesn't need any extra security.

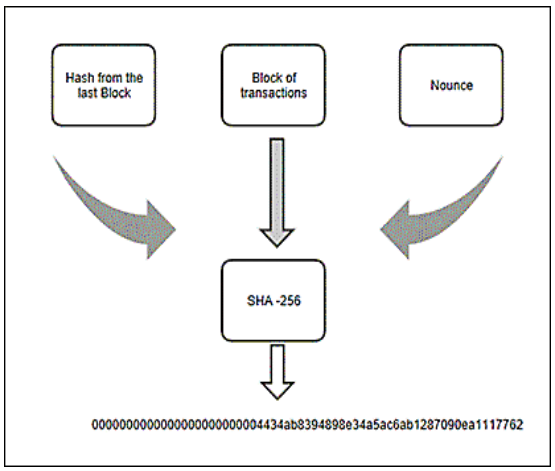


Figure 9: Decentralized Intelligence

1.5 CONVERGENCE OF BLOCKCHAIN AND DECENTRALIZED INTELLIGENCE

The convergence of blockchain with decentralized intelligence represents a trend away from monolithic, cloud-centric AI systems toward architectures where data, processing, and decision-making are spread over a trust-minimized network. While decentralized intelligence, including edge AI, federated learning, and multi-agent systems, provides adaptive reasoning, blockchain provides the layer of coordination, transparency, and motivation. Together, they construct ecosystems capable of autonomous, verifiable, and collectively managed intelligence.

1.5.1 Integrated Architecture

A data and sensing layer (IoT devices, edge nodes, and agents gathering raw data), an intelligence layer (AI/ML models carrying out inference, prediction, or decision-making, frequently dispersed across edge and agent nodes), and a blockchain coordination layer (smart contracts, consensus mechanisms, and distributed ledgers that record, validate, and settle the outputs of the intelligence layer) are the three typical layers of an integrated blockchain AI architecture. This design separates computation from control: blockchain guarantees that AI models' inputs, outputs, and interactions remain tamper-proof, auditable, and coordinated without the need for a central authority, while AI models can operate locally or across a network of agents.

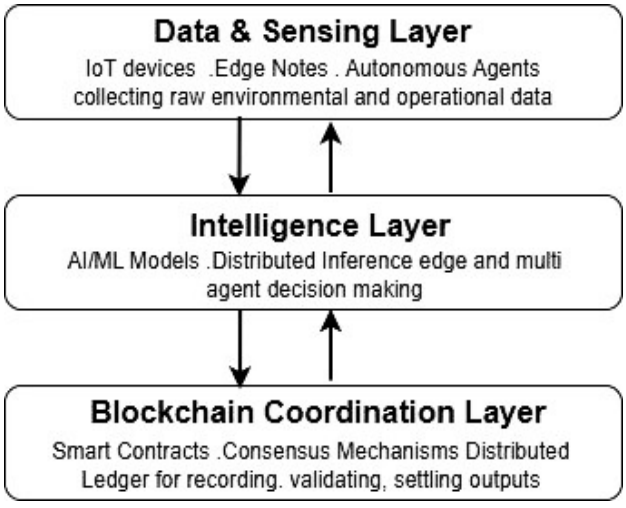


Figure 10: Integrated Blockchain of Decentralized Intelligence Architecture

1.5.2 Blockchain-Enabled AI

Blockchain-enabled AI is the application of distributed ledger technology to improve AI systems' traceability, transparency, and reliability. Verifying the integrity of training datasets, documenting model provenance (who trained a model, on what data, and when), and establishing unchangeable audit trails for AI-driven choices crucial in regulated industries like healthcare or finance are all examples of this. AI capabilities can be shared, marketed, and regulated in a decentralized marketplace rather than being under the jurisdiction of a single provider because of smart contracts' ability to automate model deployment, usage licensing, and payment for AI services.

1.5.3 Decentralized Data Governance

Because traditional AI systems rely on centralized data repositories, there are dangers associated with data ownership, abuse, and single points of failure. Decentralized data governance leverages blockchain to define transparent standards for how data is collected, accessed, shared, and monetized, providing data owners verifiable control over their contributions. Mechanisms such as data provenance tracking, consent management via smart contracts, and decentralized access rights allow various stakeholders to administer shared datasets collaboratively, assuring accountability while eliminating reliance on any single centralized custodian of data.

1.5.4 Privacy-Preserving Machine Learning

AI models can be trained across dispersed data sources without disclosing raw data by combining blockchain technology with privacy-preserving methods like federated learning, differential privacy, and secure multi-party computation. In this configuration, the underlying privacy mechanisms guarantee that sensitive data never leaves its original location, while blockchain coordinates the training process, checks the integrity of model updates supplied by various participants, and stops malicious tampering or data poisoning. This is especially helpful in fields where direct data sharing is restricted by ethical and legal issues, such as healthcare and finance.

1.5.5 Tokenized Intelligent Ecosystems

By using blockchain-based tokens to represent data, computational resources, model access, or agent services, tokenization adds financial incentives to decentralized intelligence networks. Participants whether human contributors, edge devices, or autonomous agents can be compensated for giving data, processing power, or trained models, creating self-sustaining ecosystems where intelligence is produced, transferred, and consumed through transparent, market-driven methods. Additionally, tokenized governance aligns incentives throughout a decentralized community rather than consolidating control in a single body by enabling stakeholders to vote on network parameters or resource allocation.

CHAPTER 2

DISTRIBUTED LEDGER ARCHITECTURES AND ECOSYSTEM DESIGN

Contributed by: Dr. Chayan Paul, Ankita Saha, & Mitu Roy

2.1 INTRODUCTION

Distributed ledger technology (DLT) has evolved from early cryptocurrency-focused blockchains into a diverse family of architectures underpinning complex digital ecosystems across finance, supply chains, health, and the Internet of Things (IoT). This chapter examines distributed ledger architecture and ecosystem design from a systems perspective, focusing on how consensus mechanisms, data models, network topology, governance, and tokenomics interact to shape performance, security, and sustainability. It first introduces core concepts and distinguishes between blockchain, directed acyclic graph (DAG)-based ledgers, permissioned platforms, and interoperable multi-chain systems. It then reviews related work on reference architectures, enterprise design frameworks, and application-specific DLT deployments, highlighting gaps around interoperability, post-quantum security, and long-term economic viability. The methodology section proposes an architectural analysis framework that decomposes DLT systems into layered components (network, consensus, data, execution, governance, and incentive layers) and evaluates them along criteria including scalability, decentralization, security, interoperability, and regulatory alignment. Using this framework, the chapter analyzes state-of-the-art technologies such as Ethereum's proof-of-stake rollup-centric roadmap, Hyperledger Fabric's modular permissioned architecture, Polkadot and Cosmos interoperability protocols, and DAG-based systems like IOTA for IoT. It also discusses emerging trends in post-quantum DLT, sustainability-aware consensus, and formal methods for protocol verification. The chapter concludes by outlining open research directions in ecosystem design, including cross-domain interoperability, incentive-compatible governance, and end-to-end architectures for IoT and cyber-physical systems.

2.1.1 The Genesis of Decentralized Trust

The paradigm of distributed computing underwent a monumental shift with the introduction of Byzantine fault-tolerant ledger systems operating in permissionless environments. Traditionally, distributed systems have employed trusted coordinator nodes, centralized clearinghouses, or fixed sets of validators with identity to reach consensus. The traditional systems were built using fundamental theoretical computer science knowledge developed

during the latter half of the twentieth century. They assumed that members of the distributed network could be identified and authenticated and could face legal/administrative sanctions. In a system like this, to establish agreement about the current state of the ledger, it was necessary to solve the classic Byzantine Generals Problem under the constraints of network synchronization and fault tolerance. However, such systems inherently did not possess the ability to scale up globally in an open, hostile, and untrusted environment. The combination of cryptographic hashing, public key encryption, and game theory was what ultimately shook up this paradigm. In using these seemingly independent domains to construct a coherent operational system, the design of distributed ledger technology became one in which the construction of state machines was made possible in such a way that there is a single immutable reality among a network of adversarial actors spread out across the globe. It was through the use of cryptographic hashing that this mathematical operation became possible to link these data packages in an unalterable chain.

The Cryptographic Hash Function -

$$H(x) = y \qquad (1)$$

where finding any $x' \neq x$ such that $H(x') = y$ is computationally infeasible.

Public-key cryptography established a robust, decentralized identity paradigm, allowing users to execute state transitions and sign transactions securely without revealing their private keys or relying on a centralized identity provider. Crucially, the integration of economic game theory bridged the gap between theoretical fault tolerance and practical network security. By introducing algorithmic incentives, such as minting native tokens and distributing transaction fees, protocols turned potential attackers into rational economic actors who found it more profitable to support the network's consensus than to disrupt it. This synthesis transformed distributed systems from passive data repositories into dynamic, self-sustaining state machines capable of preserving deterministic integrity in the presence of malicious Byzantine actors.

Byzantine Fault Tolerance Condition

$$n \geq 3f + 1 \qquad (2)$$

In a network of n total nodes, a Byzantine fault-tolerant (BFT) system can safely tolerate up to f faulty (malicious or crashed) nodes only if the total number of nodes satisfies $n \geq 3f + 1$. Equivalently, $f \leq (n - 1) / 3$. This guarantees that honest nodes always form a strict two-thirds supermajority, so the network can still agree on a single, correct value even when faulty nodes actively lie.

2.1.2 Evolution from Simple Ledgers to Programmable State Machines

As the domain matures, the focus of academic inquiry and industrial application has pivoted from basic transactional ledgers to complex, multi-layered architectural frameworks capable of executing arbitrary computational logic via smart contracts. The earliest iterations of distributed ledgers were structurally constrained, designed explicitly to serve as single-purpose peer-to-peer electronic cash systems. These early frameworks utilized restricted, non-Turing-complete scripting languages to minimize the attack surface and ensure predictable transaction execution.

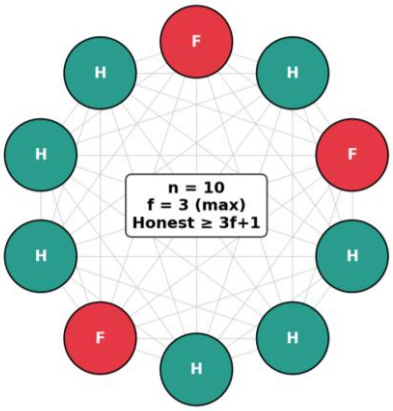


Figure 1: Byzantine Fault Tolerance - honest (H) vs. faulty (F) nodes; the system stays safe while $n \geq 3f + 1$.

While highly secure for simple value transfers, these primitive architectures lacked the expressive capacity required to implement sophisticated conditional logic, automated escrow systems, or decentralized financial primitives. The realization that a distributed ledger could serve as a generalized, programmable virtual machine catalyzed a profound architectural evolution. By embedding a deterministic, Turing-complete execution engine - such as the Ethereum Virtual Machine (EVM) or WebAssembly (WASM) runtimes - directly into the state transition logic of the ledger, developers were empowered to deploy immutable code that executes exactly as written. This transformation altered the fundamental nature of the ledger: it ceased to be a passive record of historical balances and became an active, global computational layer. Transactions were no longer merely transfers of currency; they became state-changing inputs that triggered complex algorithmic logic across a distributed network of nodes. This programmable paradigm gave rise to the decentralized application (dApp) ecosystem, enabling the creation of autonomous organizations, algorithmic lending protocols, and tokenized asset structures. However, this vast expansion of computational utility introduced severe strains on the underlying protocol layers. The requirement that every node in the network sequentially execute every line of smart contract code to maintain state synchronization. created a massive computational bottleneck, exposing the systemic vulnerabilities inherent in early monolithic ledger designs.

2.1.3 The Core Structural Bottleneck: The Blockchain Trilemma

This evolution has exposed the fundamental limitations of early ledger designs, most notably encapsulated in the blockchain trilemma, which asserts that a decentralized ledger can simultaneously achieve only two of three properties: security, scalability, and decentralization. Security dictates the network's resilience against malicious takeover, code exploitation, and structural attacks, requiring significant computational or economic capital to alter historical states. Scalability measures the network's capacity to process an increasing volume of transactions efficiently, quantified by throughput metrics such as transactions per second (TPS) and time-to-finality (TTF). Decentralization defines the distribution of validating power and infrastructure across an expansive, diverse set of independent operators, ensuring that the network remains permissionless and resistant to censorship. In a monolithic system architecture, these three vectors exist in a state of permanent tension. To maximize scalability,

a network might be tempted to restrict the number of validating nodes or increase the hardware requirements for node operators, thereby accelerating block propagation and transaction validation speeds. However, this strategy directly undermines decentralization by concentrating validating authority into a small cartel of well-capitalized data centers, rendering the system vulnerable to regulatory capture or collusive attacks. Conversely, if a network prioritizes radical decentralization by keeping node hardware requirements low enough for consumer-grade devices, the network's throughput becomes strictly bounded by the bandwidth and processing constraints of the slowest participating nodes.

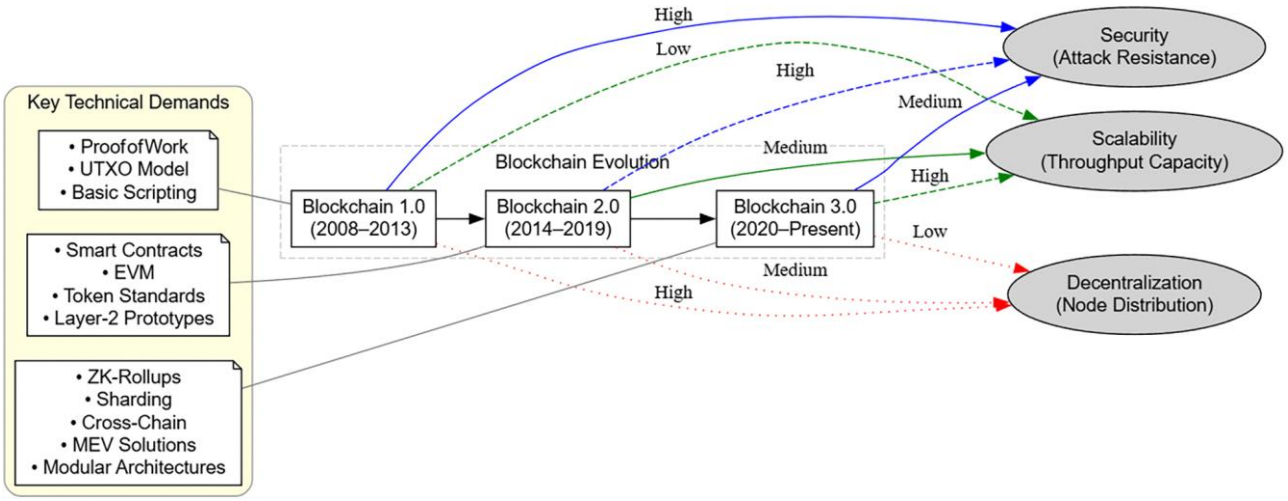


Figure 2: Blockchain generations and trilemma pressure intensification

Furthermore, attempts to increase throughput by expanding block sizes or accelerating block production intervals without addressing these foundational constraints lead to high block-orphan rates and severe network degradation, compromising overall security. Consequently, navigating this trilemma stands as the central engineering challenge of modern distributed systems design, forcing researchers to look beyond simplistic optimizations toward profound structural re-engineering.

$$\boldsymbol{Transaction\ Throughput\ (TPS)} = \frac{BlockSize_{transactions}}{BlockTime_{(second)}} \quad (3)$$

Throughput, measured in transactions per second (TPS), is the number of transactions a block contains divided by the time taken to produce that block. This single ratio explains the trilemma trade-off in numbers: a network can raise TPS by packing more transactions per block or by shortening block time - but both changes increase the bandwidth and storage burden on every validating node, which pushes out smaller, less powerful participants and reduces decentralization.

2.2. ARCHITECTURAL INNOVATION: DATA, CONSENSUS, AND TOPOLOGY

Modern distributed ledger architectures attempt to resolve these trade-offs through innovative modifications to the data structure, consensus layer, and network topology. At the data structure level, the industry is transitioning away from rigid, linear block sequences toward parallelized data models such as Directed Acyclic Graphs (DAGs) and state-sharding mechanisms. Linear structures force a total, sequential ordering of all transactions, creating artificial queuing delays

even for entirely unrelated transfers. By adopting DAG structures, protocols can process transactions asynchronously and concurrently, treating individual transactions as independent vertices that validate prior actions, thereby decoupling throughput from block timing constraints.

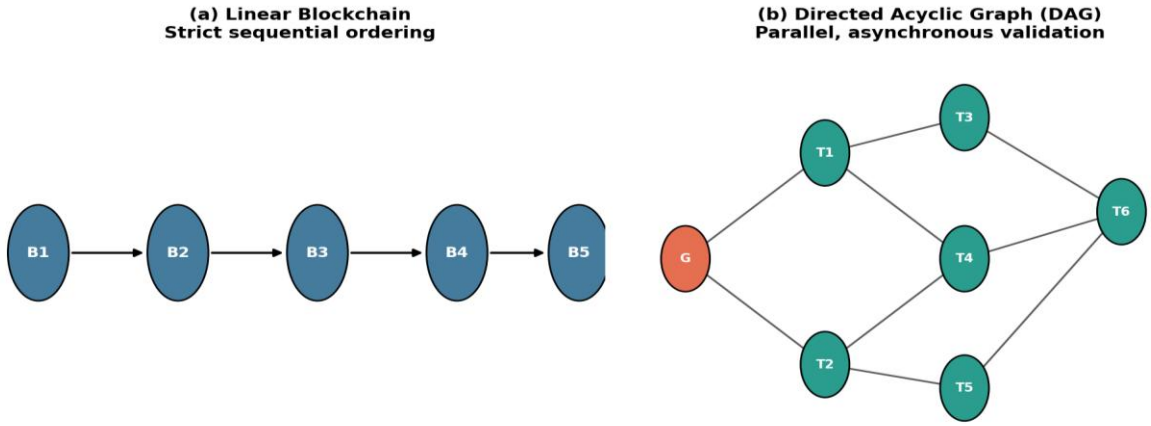


Figure 3: Linear blockchain vs. DAG-based ledger structure - DAGs allow many transactions to be validated in parallel instead of one strict queue.

State sharding further revolutionizes the data layer by partitioning the global ledger state into multiple distinct, parallel sub-networks, or shards. Each shard processes its own subset of transactions and maintains its own state history, multiplying the cumulative throughput of the network horizontally as more nodes join the ecosystem.

Throughput Scaling Under Sharding

$$〖TPS〗_{total} = k \times 〖TPS〗_{shard} \quad (4)$$

Nonetheless, this idea faces challenges due to the need for communication between the different shards. However, there have been advancements in the consensus layer whereby the inefficiency of the PoW consensus mechanism has been improved to an efficient one, PoS, PoA and BFT mechanisms. These new consensus mechanisms ensure that the block proposal stage and the block confirmation stages are separated. Through the use of cryptography sortition and pseudorandom validator selection mechanisms, these protocols can select small, highly secure committees to validate and finalize state transitions in sub-second intervals. This structural bifurcation allows networks to achieve deterministic finality - where a transaction is guaranteed to be irreversible immediately upon validation - in stark contrast to the probabilistic finality models of early PoW systems that required users to wait for multiple subsequent blocks to assure transaction security. In Proof-of-Stake, the probability that validator is chosen to propose the next block is proportional to the fraction of total staked tokens that validator controls: $(stake\ of\ validator\ i) \div (sum\ of\ all\ validators'\ stake)$. This replaces energy-intensive computation with capital-at-risk as the scarce resource that Sybil-proofs the network.

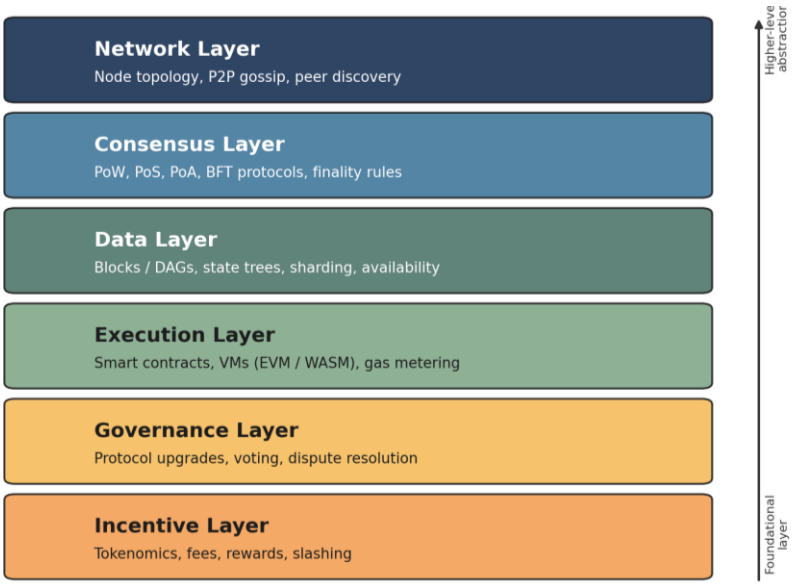


Figure 4: The six-layer Distributed Ledger Architecture stack - network, consensus, data, execution, governance, and incentive layers.

Furthermore, alterations in network topology, such as the emergence of modular blockchain designs, isolate execution, settlement, consensus, and data availability into specialized, dedicated layers. Rather than forcing a single layer to bear the full weight of the entire computational cycle, modular topologies offload execution to high-performance layer-two rollups while relying on a highly decentralized layer-one foundation exclusively for consensus and immutable data availability.

2.3 THE SOCIO-ECONOMIC ARCHITECTURE: ECOSYSTEM DESIGN

Simultaneously, the engineering of these platforms requires careful consideration of ecosystem design, which encompasses the economic incentives, governance structures, and interoperability protocols that ensure the long-term sustainability and utility of the network. A technically flawless ledger architecture will ultimately fail if its underlying economic model fails to foster sustained alignment among its diverse constituents, including developers, validators, liquidity providers, and end-users. Tokenomics serves as the economic bedrock of the ecosystem, dictating the rules of asset issuance, inflation management, and transaction fee distribution. Modern ecosystem design employs algorithmic fee-burning mechanics and dynamic subsidy regimes to create a balance between inflationary security rewards and deflationary utility forces, aligning the long-term value of the native asset directly with the transactional velocity and adoption of the network.

Net Token Supply Change (Issuance vs. Fee-Burn)

$$\Delta S = R_{issued} - R_{burned} \quad (6)$$

The change in circulating token supply over a period equals the newly issued rewards (paid to validators/miners) minus the tokens permanently removed through fee-burning mechanisms. When $\Delta S > 0$ the token is net inflationary; when $\Delta S < 0$ it is net deflationary. Beyond pure

economics, the governance structures embedded within the ecosystem dictate how the protocol adapts to systemic shocks, technical vulnerabilities, and evolving market demands. Ecosystem designers must balance the predictability of immutable code with the flexibility required to upgrade complex software systems over multi-year lifecycles. This has driven intense research into formal on-chain governance mechanisms governed by tokenized voting primitives, as well as decentralized off-chain improvement frameworks that mirror traditional open-source software governance but are reinforced by cryptographic checks and balances.

Finally, as the landscape fragments into a multitude of heterogeneous, specialized networks, the inclusion of robust interoperability protocols has become an existential necessity for ecosystem viability. Without secure, trust-minimized cross-chain communication vectors, assets and data become stranded in isolated silos, limiting network effects. Modern ecosystem engineering therefore treats cross-chain bridging primitives, light-client relays, and universal messaging standards not as external add-ons, but as core architectural pillars designed to facilitate seamless, secure interactions across the global web of decentralized ledgers. A substantial body of literature surveys distributed ledger technologies from various perspectives, including consensus protocols, application domains, security, and performance. IoT surveys highlight the role of DLTs in offering decentralized trust, auditing capabilities, and device identity management in applications such as smart homes, cars, and smart cities; however, there are concerns regarding resource constraints and latency issues. In other studies, researchers investigate the application of DLT in providing networked security, including authentication and access control based on identities and smart contracts. An emerging body of research focuses on post-quantum distributed ledgers since there is the realization that a number of the public key cryptography approaches that have been applied in today's blockchains can be easily compromised by quantum computers. Literature surveys on this subject classify various kinds of quantum resistant signature algorithms, proof systems, and hash functions, and evaluate their viability for use in DLT based on the size of keys required, complexity of verification, and compatibility with existing infrastructure.

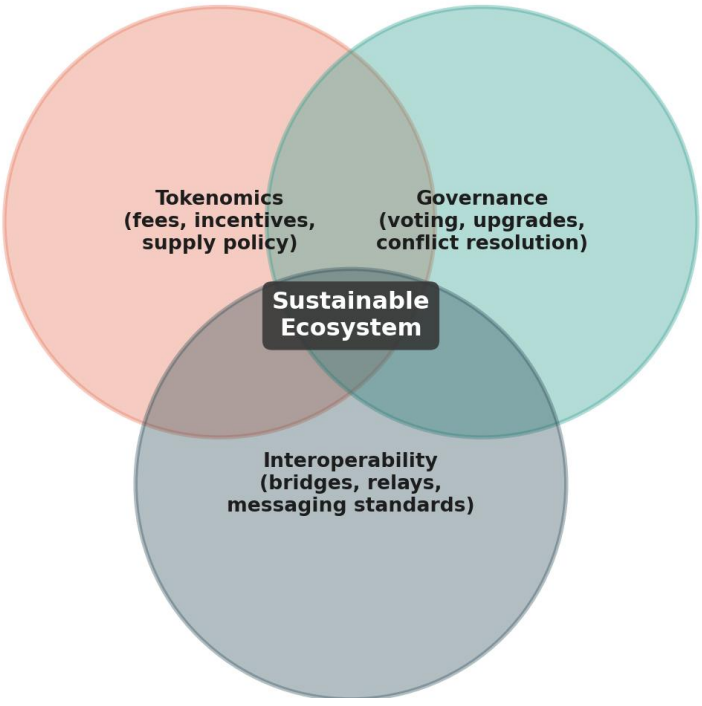


Figure 5: Ecosystem Design - tokenomics, governance, and interoperability jointly determine ecosystem sustainability.

Despite all of these advances, there is still a significant number of surveys that have a domain-based or technology-oriented focus, such as IoT and security, as opposed to providing an eco-system design perspective that relates protocols to governance and regulation and to tokenomics. Hence, the integrative approach considering DLT technologies from the perspective of layered socio-technical systems is needed.

2.5. FRAMEWORKS FOR ENTERPRISE AND ECOSYSTEM DESIGN

The use of blockchain and DLT technology within enterprises has prompted the development of various design frameworks and best practices regarding the design of the technology within organizational contexts. One such framework that can be mentioned here is the Enterprise Blockchain Design Framework (EBDF). Some of the key decisions made using this framework include permissioning options, data protection and privacy concerns, logic of smart contracts, and compatibility of the system with existing infrastructure.

Best practice studies of DLT usage in financial market infrastructure, custodian services, and post-trade services by industry bodies include the experiences learned from practical deployment of the technology. This includes experiences in participant onboarding, lifecycle management of digital assets, interoperability with existing payment and messaging solutions, and compliance with regulations. It is evident that well-governed risk management and well-defined data model and interface standards are just as important as good technological performance in an ecosystem. Additional research studies have examined token and incentive designs as vital components of blockchains. Tokenomic frameworks that adopt strategic approaches highlight the importance of defining the purpose and distribution mechanisms of tokens in addition to ensuring proper alignment of incentives among users, validators, and developers. Such research claims that ill-conceived economic designs may result in unsustainability or susceptibility to collusion despite a strong technical infrastructure.

2.5.1 Reference Architectures for System Interoperability

The goal of reference architectures in DLTs is to define reusable abstraction and design pattern for commonly used components like consensus layer, smart contract engine, and client interface. A number of academic initiatives and collaborations have suggested layering of the system in terms of network layer, consensus layer, data layer, and application layer through plug-in interfaces that allow for diverse consensus and execution engines. The problem of interoperability is among the most important topics because of interaction between heterogeneous ledgers. Specific research devoted to the issue of interoperable distributed ledger technology, involving cataloguing techniques like atomic swaps, sidechains, relay chains, and hub-and-zone architecture. In this context, IOTA and other similar researches focus both on studying of DLTs appropriate for use in IoT and on more general issues of interoperability. There are some blockchain projects like Polkadot and Cosmos that seek to build multi-chain ecosystems where there is a central relay/hub chain that acts as a framework for diverse application-specific chains. However, the research on DLT technology applications in the case of

identification and access control raises the question of standardization of identity models and trust anchors for various types of environments, especially federated and/or multi-cloud environment. Thus, the above research proves that the adoption of interoperable DLT technology is coming together with the standards of identity and access management.

2.5.2. Challenges and Opportunities for Future Research

The surveys, frameworks, and reference architectures taken together give us a lot of information about certain facets of DLT; however, there are still some gaps left when one looks at these works from the perspective of the ecosystem approach. First of all, many researchers pay their attention to such aspects as consensus and data structures but do not take into account governance, incentives, and regulation, which are the key elements making the whole ecosystem workable. Second, the problems associated with interoperability of different networks were examined only through the prism of the technical connection between them. Third, although the topic of post-quantum security becomes more and more relevant concerning blockchain technologies, the issue of transformation of existing blockchains into quantum-safe technology has not been analyzed yet. Fourth, even though there are many innovative ways of addressing the problem of sustainability both technically and economically, the problem itself has not been solved yet.

2.6. ARCHITECTURAL ASSESSMENT FRAMEWORK

In order to undertake structured research about the architectural and ecological aspects of the DLT, the following framework has been considered that has been created based on the reference architecture and research roadmap of DLTs. In particular, the layers of DLT have been classified into six broad categories, which are: Network Layer, Consensus Layer, Data Layer, Execution Layer, Governance Layer, and Incentive Layer. The network layer consists of the topology of the nodes, the protocol of node connectivity, peer discovery, and bandwidth/latency optimization. The consensus layer consists of the consensus protocol (whether it be Proof of Work, Proof of Stake or BFT algorithms) and the parameters of this consensus protocol like block time and adversarial assumptions. The data layer consists of the ledger model along with the block/DAG structure and transaction formats. The Execution Layer comprises the smart contracts, the virtual machines, and the transaction pipes which have programming, determinism, and upgradability. The Governance Layer is concerned with the decision-making process in case of parameter, protocol, and dispute resolutions which can be on-chain or off-chain. The Incentives Layer comprises the token economy.

2.6.2 Evaluation Criteria

Using this layered model, architectures and ecosystems are evaluated along multiple criteria to facilitate comparative analysis. Scalability captures throughput, latency, and resource efficiency under realistic workloads. Decentralization reflects the distribution of power over consensus, governance, and resource provisioning, including metrics such as validator concentration or mining centralization. Security encompasses resistance to double-spending, reorganization attacks, Sybil attacks, and cryptographic vulnerabilities, as well as robustness under partial network partitions. Interoperability measures the ability of a ledger to exchange data and value

with other ledgers and external systems, whether through standardized interfaces, cross-chain protocols, or shared security models. Privacy and confidentiality account for protections such as encryption, zero-knowledge proofs, and selective disclosure, especially in permissioned and enterprise settings. Regulatory and compliance alignment evaluates how the ecosystem supports identity, auditability, and controls needed to comply with sector-specific regulations, particularly in finance and healthcare. Sustainability is considered in both environmental and economic terms, assessing energy consumption of consensus mechanisms and the long-term viability of token and fee models. Finally, developer and user experience capture tooling, documentation, and support for secure application development and onboarding, which strongly influence ecosystem growth but are often underemphasized in purely technical analyses.

2.6.3 Analytical Approach

The analytical approach taken in this chapter is qualitative and comparative rather than empirical benchmarking. The goal is not to provide performance measurements under uniform workloads, but to examine architectural choices and ecosystem design patterns that drive observable properties. Representative platforms are selected to illustrate different design points in the space of distributed ledger architectures, focusing on widely adopted technologies and prominent emerging approaches. For each platform or technology family, the analysis maps its design to the six-layer architecture and the evaluation criteria described above. This reveals how specific choices at the network, consensus, data, and execution layers are intertwined with governance and incentives to create an ecosystem trajectory. The analysis draws on published research, industry whitepapers, and best-practice reports, with IEEE-style citations used to attribute concepts and data where appropriate. While the method is qualitative, it aims to be systematic and transparent, enabling other researchers to extend the framework with quantitative metrics and domain-specific refinements.

2.6.4 State of the Art Technologies

State-of-the-Art Technology are the most advanced, innovative, and high-performance technologies available at a given time, developed through continuous research and innovation to improve automation, efficiency, security, and intelligent decision-making. These technologies include Artificial Intelligence State of the Art Technologies:-(AI), Machine Learning (ML), Deep Learning (DL), Internet of Things (IoT), Blockchain, Cloud Computing, Edge Computing, 5G Communication, Quantum Computing, Generative AI, Cybersecurity, and Big Data Analytics. Their operation is often represented using mathematical models, such as the AI prediction function $Y=f(X)$, the Machine Learning model $y=wx+b$, the IoT data transmission rate $R=D/T$, the Blockchain hash function $H=h(B)$, and the 5G channel capacity equation $C=Blog2(1+SNR)$. The technologies mentioned above facilitate real time processing of data, automation, secure communication and fast connection in industries such as health care, education, banking, manufacturing, agriculture and smart cities. Though there are various limitations such as cost, security and availability of knowledgeable professionals, State-of-the-Art Technologies will always keep playing an important role in digital transformation.

2.7. PUBLIC PLATFORMS FOR SMART CONTRACT EXECUTION

Smart Contracts that do not Require Permission. Some platforms include Ethereum, Solana, BNB Chain, and Avalanche, where there is a possibility of developing smart contracts in a decentralized and permissionless computer system. The smart contract is referred to as an automatic agreement and performs some specific tasks depending on particular situations. Using such platforms helps in creating DeFi, NFTs, games, DA-pps, and identity systems. A basic example of executing a self-executing contract is (O = f(C, I)), where (O) is the output of the code, (C) is the contract code, and (I) is the input data. Smart Contract Platforms are foundational to the rapidly expanding deployments of modern applications on Web3 and Blockchain technology. With cryptography, the data within a block is ensured with integrity (H = h(B)), with (H) being the hash value and (B) being the block.

2.7.1. Rollup-Centric Strategy for Ethereum Scaling

Ethereum is no longer the proof-of-work chain that it used to be; today's Ethereum is a proof-of-stake blockchain focusing on scalability via rollups and sharding of data availability. The consensus layer in Ethereum has changed to proof-of-stake, where validators stake native tokens to create and validate blocks and probabilistically reach finality through checkpointing and justification techniques. There is a chain which is said to be canonical with respect to transactions and state data, and roll-ups will feed transaction data into Ethereum blockchain in the data layer of Ethereum. Operationally, the blockchain of Ethereum facilitates the building of smart contracts using Ethereum Virtual Machine (EVM) with gas through deterministic execution of bytecode. The process of governance on Ethereum blockchain is off-chain where Ethereum Improvement Proposals (EIPs) are discussed in the community while implementing them in clients even if there are few researches on governance on chain. On the other hand, in terms of incentive design, there is block reward, transaction fees, fee burning, among others that will influence token supply.

2.7.2 Comparative Public Blockchain Architectures

In addition to Ethereum, there are several open-source DLT platforms developed within the corresponding segment of the design space, and their objective could be related to the increase in performance, development of a new consensus algorithm, or implementation of new data structures. Thus, on the one side, certain DLT platforms use proof of stake finality algorithms or even delegated proof of stake, while, on the other side, there is another group of DLT platforms which implement hybrid solution combining BFT consensus and sharding of the execution. All the aforementioned ecosystem developed specifically for this segment of the design space with regard to governance and incentive mechanisms could be employed for resource allocation across chains and security design. Particularly, in relation to relay chains and hub chains, certain approach to the assignment of roles and stake and slashing rules might be used.

2.7.3 Enterprise-Grade and Consortium Distributed Ledger Platforms

Enterprise Platform or Permissioned Platforms are known as those blockchain platforms that provide secure and private means of accessing the blockchain network through providing access only to certain people. While in the case of public blockchain, there are no restrictions on access, the access is restricted to the members in the case of permissioned platform who can validate the transaction and access the data in the blockchain. The examples of some popular enterprise platforms are Hyperledger Fabric, R3 Corda, and Quorum. The integrity of

transactions is ensured using a cryptographic hash function, H = h(B), where H is the hash value and B is the block data. Data integrity can also be verified by checking H_new = H_stored, indicating that the data has not been altered. This has proven to be a very successful method in dealing with fast and low latency transactions. This method is ideal for those who need to transfer data in a secure way.

2.7.4. Modularity in Permissioned Blockchains: A Hyperledger Fabric Case Study

The example of such modularity in DLT architectures is provided by Hyperledger Fabric. To begin with, when it comes to the network and consensus layers of the Hyperledger Fabric, it can be pointed out that the processes of transaction endorsement and ordering are separated. In other words, organizations can develop their own policies for endorsement and employ different orders. In regard to the Execution Layer, it should be noted that the transactions can take place in the Fabric chaincode, in which case the container can be coded using any programming language. The drawback of the above system is that it is very difficult to make the above process deterministic. Regarding the governance aspect, it will be related to the membership service provider and organizations. Thus, there will be a chance to connect the transactions executed through the blockchain platform with real-world entities. It is necessary to note that the motivation for permissioned systems is not connected with the open token market but with the benefits of collaboration.

2.7.5. Industry-Specific Blockchain Deployments

The financial sector uses permissioned DLT technology to perform post-trade activities, to handle collateral management activities, and tokenization of securities, typically with focus on interoperability with existing payment infrastructure. Best practices document talks about architecture, where there is a correlation between asset ledger and messaging and custody system, where there is a body responsible for standardization and change management. In the same manner, there is also a need for private data sharing and auditing in healthcare and logistics industries, hence making use of permissioned ledgers and off-ledger solutions architecture. Regarding the matter of network access control and identification, DLTs can handle issues relating to distributed identities and the policies of their authorization. In fact, some research works were carried out to classify such systems depending on the manner of implementing access control on/off chain.

2.7.6. IoT and Cyber-Physical Systems

The convergence of DLT and IoT has been widely discussed in the literature, with surveys documenting use cases from smart homes and smart cities to autonomous vehicles and industrial automation. DLT may help achieve security through logging, device identification, and decentralization coordination; however, it faces limitations in terms of bandwidth, power consumption, and latency. Patterns may include hierarchical DLT, which uses light weight ledgers maintained by edge or local networks that sync with global ledgers periodically. While designing an ecosystem for IoT through DLT, it is necessary to consider the interplay between various layers such as edge computing, network function virtualization, and cloud platform. The governance can have different participants such as device manufacturers, service providers, regulators and more, which makes it essential to have some guidelines for updating software, managing keys, and determining liabilities. An incentive mechanism should award devices with

rewards for providing data/services and penalties for wrongdoing but must be simple and straightforward.

2.8. INTEROPERABLE ARCHITECTURES ACROSS BLOCKCHAIN NETWORKS

The concepts of interoperability and Cross-chain ecosystems have been developed in such a way that different blockchain platforms can communicate and transfer information, including digital currencies, without any problems while not relying on any centralized entity. In this case, different blockchains can co-exist. The most widely used interoperable blockchains include Polkadot, Cosmos, and Chain-link. Cross-chain communication can be represented as $C = A \leftrightarrow B$, where A and B are two different blockchain networks exchanging data or assets. Blockchain interoperability enables the tearing down of network barriers, the effective use of shared resources, increased efficiency in transactions, and serves as the basis for developing scalable blockchain solutions.

2.8.1. Architectural Patterns for Cross-Chain Interoperability

In many respects, the creation of technologies, which enable various ledgers to operate in conjunction, is an evolving process owing to the presence of numerous applications that require the use of various chains in order to form their own ecosystems. Atomic swap is among such patterns that enable transactions to take place across chains without depending on any trust-based mechanism whatsoever. These security features include the non-existence of double spending across multiple chains and state consistency across multiple chains. There are other issues related to interoperability including governance that comes about with the use of bridges, coordinated upgrades and risks sharing across multiple chains. In this ecosystem design, it is not only the governance of one ledger that should be considered but several domains.

2.8.2 Economic and Regulatory Implications of Blockchain Technology

There are many economic problems that arise in inter ledger systems in relation to other associated chains, such as value capture, congestion charges, and incentive problems. The tokenomics should take into consideration the problem of usage and fee leakage between one chain and the other because it can be highly detrimental to the security or governance of the lesser-used chain. In relation to regulations, the problem is made even more complicated where there is asset movement between one chain and another. Guidelines within finance also reveal the necessity of conducting proper risk assessment, treating interoperation partners with respect, and delegating accountability with regards to cross-chain collaboration. It becomes clear, therefore, that it is now imperative to acknowledge that interoperation should not be viewed only as a technical issue but also as a legal one.

2.9. POST-QUANTUM AND SECURITY-ENHANCED BLOCKCHAIN ARCHITECTURES

Post-Quantum and Security Aware Architecture refers to complex architectures of computer and blockchain security which seeks to provide protection for computer networks from any attack done by using conventional as well as future quantum computers. Post-Quantum and

Security Aware Architecture utilizes post-quantum cryptography, key management, digital signatures, and layered security techniques. A basic encryption model is represented as C = E(K,P), where P is the plaintext, K is the encryption key, and C is the ciphertext, while decryption is P = D(K,C). In quantum cryptography, the state of a qubit is expressed as |ψ≥=α|0>+ β|1> with the condition $〚|\alpha|〛$ ^2 + $〚|\beta|〛$ ^2 = 1. By combining quantum-resistant encryption with security-aware system design, these architectures provide stronger protection against emerging cyber threats, ensuring secure communication, data privacy, and long-term resilience in next-generation digital ecosystems.

2.9.1 Quantum Threats and DLT

The emergence of quantum computing threatens many cryptographic primitives relied upon by existing blockchain and DLT systems, particularly those based on discrete logarithms and factoring. Systematic surveys of post-quantum distributed ledger technology analyze candidate quantum-resistant signature schemes and their implications for transaction sizes, verification costs, and key management. These works show that naive substitution of quantum-safe primitives can significantly impact performance and storage requirements, suggesting that architectural adaptations may be necessary. In light of ecological considerations, moving towards post-quantum cryptography poses issues in terms of governance and coordination. Large-scale systems will have to come up with common rules and approaches towards implementing post-quantum crypto, while performing updates and key rotation. In order not to fragment ecological systems, researchers emphasize the importance of creating ledgers with the possibility of being upgraded and flexible regarding cryptography.

2.9.2. Comprehensive Security Frameworks and Formal Verification

Apart from quantum security aspects, the holistic approach to security for DLTs includes the formal verification of the consensus algorithms, smart contracts, and interoperability protocols. Research agendas in DLTs require careful performance and security analysis taking into consideration the delays in the network, partial synchrony, and possible adversarial actions. Formal verification techniques can help find hidden bugs in the protocol design. Security is also considered at the socio-technical level, where mechanisms for governance are employed to guard against risks associated with centralization, conspiracy, or capture. It is critical that tokenomics and governance models are crafted in a way that deters certain attacks like censorship, bribery, and the acquisition of validation power by a majority. From this perspective, strong distributed ledger designs are those that encompass all forms of security together.

CHAPTER 3

CRYPTOGRAPHIC PRINCIPLES FOR SECURE COMPUTATION

Contributed by: Dr. Pabitra Pal, Soumyajit Jalua, Biplab Bhunia

s

3.1. INTRODUCTION AND FOUNDATIONAL SECURITY PRINCIPLES

The emergence of blockchain technology and artificial intelligence as a pair has created new terms for negotiating trust, computing, and data exchange among autonomous and semi-autonomous agents. Underpinning such a convergence is cryptography, the branch of science whose guarantees underlie decentralized technologies in the absence of central authority. In contrast to traditional information systems relying on trust, banks, certificate authorities and centralized administrators, decentralized and intelligent ecosystems replace this trust with provable mathematical guarantees. There are several numbers of interconnected issues stemming from the substitution of trust with cryptography. First, data needs to be secured both at rest and in transit. Secondly, authentication of participants anonymous identity is required. Thirdly, computations need to be done over encrypted or distributed data, not over centrally stored one. And fourth, agreement protocols allowing distrustful nodes to reach consensus on a state need to be based on cryptographic tools. With increasingly sensitive applications being deployed by intelligent systems - ranging from medical record processing to behavioral telemetry, cryptography moves from a secondary aspect of security to the core constraint. This chapter discusses cryptographic primitives and more advanced constructs underlying cryptographic layer of such systems.

3.1.1. Confidentiality, Integrity, Authenticity, and Non-Repudiation

Before delving into particular algorithms, it is important to settle the terminology used to gauge their claims of security. The four basic features on which modern cryptography operates can be succinctly termed as the CIA plus non-repudiation paradigm. Confidentiality refers to making sure that the information is available to authorized individuals alone, and is normally secured using encryption. Data integrity refers to ensuring that the data remains unaltered, either accidentally or intentionally, ever since it was made or last verified, and is normally secured using hash functions and message authentication codes. Authenticity refers to identifying the source of the information or the one who created it or sent it, and is normally done using digital

signatures. Non-repudiation makes sure that the party cannot deny ever carrying out a particular act, like signing a transaction.

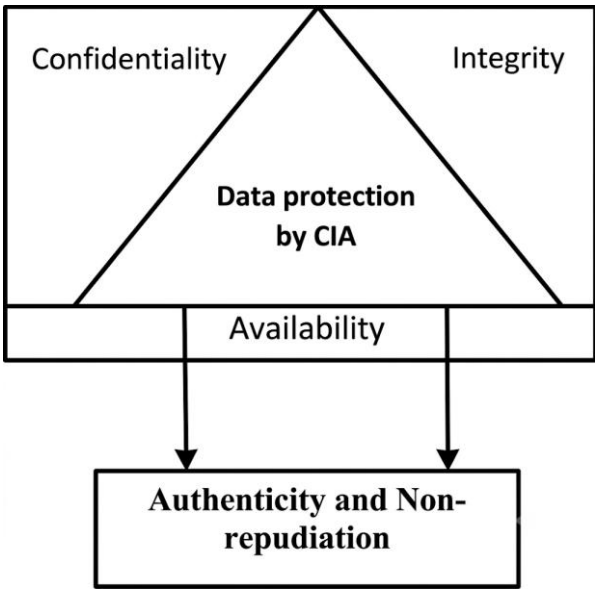


Fig 1: Data Protection

The significance of these four characteristics increases when there is no central administrator able to verify them ex-post. A blockchain transaction must be inherently verifiable and tamper-proof during that period it is verified since there will be no subsequent ability to refer back to some institutional record. The left of the chapter can be viewed as an answer to the following question: what techniques allow distrustful parties in a network to meet all four characteristics? An example can be used to demonstrate how these four properties interact with each other. Consider a case where an individual using a blockchain network intends to send a virtual asset to another individual. Confidentiality is not absolute since the amounts of the transaction and the addresses involved are visible to everyone who inspects the ledger due to the requirement of public verifiability at the expense of a central verifier, although the true identity associated with the addresses is not known but remains pseudonymous. Integrity is assured once the transaction is propagated to the network: any tampering to the transaction details will alter its hash function and can easily be detected by any node that recalculates the hash function. Authenticity derives from the digital signature associated with the transaction because only the owner of the private key used to sign the transaction would have the ability to produce such a signature. Non-repudiation arises directly from authenticity because once the digital signature is tied to the contents of the transaction and a private key through mathematical processes, it becomes difficult for the signer to deny the transaction.

3.1.2. Computational Hardness Assumptions

The above guarantees are not absolute; they depend on computational hardness, which posits that certain problems can be solved in theory demand excessive computational resources for it so within any feasible period of time. There are three kinds of hardness assumptions, with a fourth one growing increasingly popular. The integer factorization hardness assumption states that factoring a multiplication of two enormous prime numbers is computationally infeasible despite the ease of multiplication of these numbers and is the basis for the RSA algorithm presented in section 3.2. Discrete logarithm hardness assumption postulates that, given a cyclic

group and elements g and gx, recovering x is computationally infeasible and is used for classical Diffie-Hellman key exchange and El-Gamal cryptosystem. An adaptation of this assumption for elliptic curves known as elliptic curve discrete logarithm problem (ECDLP) provides the basis for elliptic curve cryptography currently used in blockchains because of the favorable security-to-key size ratio. Another family of hardness assumptions has become increasingly important as quantum computers have progressed; the Learning With Errors (LWE) and Short Integer Solution (SIS) problems provide the hardness assumptions underlying the post-quantum cryptosystems considered in Section 3.5. The effect of making one choice or another in terms of hardness assumptions is quite practical; a blockchain network that utilizes RSA for its digital signatures, for example, would have key sizes substantially greater than the ones using elliptic curves for the same level of security. Each extra byte in the key or the resulting signature is an extra byte which all full nodes of the network have to store, send, and verify for each transaction ever made in the blockchain. That is one of the reasons behind the popularity of elliptic curve cryptography even though it is based on a relatively young mathematical assumption compared to RSA it just gives you more security per byte.

3.2. SYMMETRIC AND ASYMMETRIC CRYPTOGRAPHIC PRIMITIVES

3.2.1. Block Ciphers, Stream Ciphers, and Modes of Operation

Given the identified threat environment, the chapter now focuses on the tools that help in dealing with the threat, starting with the oldest and least computationally expensive tool: symmetric key cryptography, where a single shared secret key is used for the purpose of encryption and decryption. These can be classified into two main categories symmetric cryptosystems, known as block ciphers and stream ciphers. A block cipher is an encryption method which encrypts data blocks, either 64 or 128 bits, in each cycle. Block ciphers employ consecutive cycles of substitution and permutations on blocks of plaintext to produce ciphertext blocks, and all these rounds are done under a single key. However, since block ciphers have deterministic output per block, they are generally utilized, together with a mode of operation, for example, ECB, CBC, CTR, GCM, etc., so that process messages containing more than one block, introducing some element of randomness into the process (using IVs). Examples of popular block ciphers include DES, AES, Blowfish, and 3DES. The block ciphers use fixed block size and have several rounds of confusion/diffusion; they are comparatively slower per byte compared to stream ciphers but good for structured data and storage encryption.

In a stream cipher, the process of encryption is done in such a way that data is encrypted either on a bit-by-bit basis or byte-by-byte basis. The stream cipher generates a keystream that is pseudorandom and uses the XOR function to encrypt it with the plaintext. This is a type of encryption where there is no block size at all; this means that it can continue the process of encryption and decryption until the message has been sent through. Some of the basic types of stream ciphers include RC4, Salsa20/ChaCha20, SNOW-V, and A5/1.

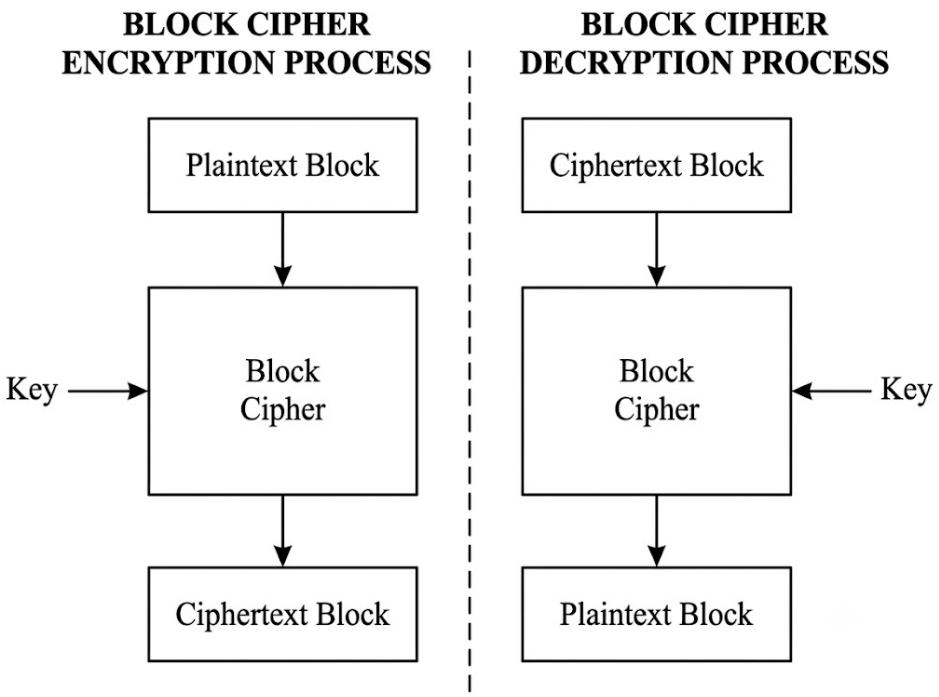


Fig 2: Block Cipher Basic Flowchart

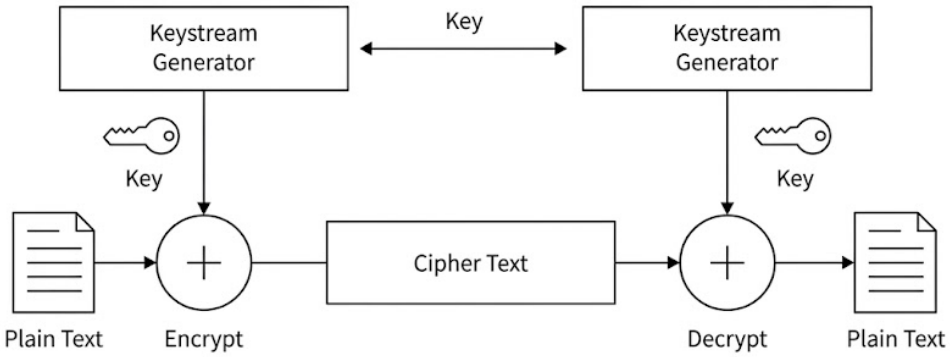


Fig 3: Stream Ciphers Basic Flowchart

The global benchmark for data security, AES prevalent block cipher currently in use. It processes of data segments of 128 bits utilizing key lengths of 128, 192 or, 256 bit keys, and is specified by NIST as an upgrade from the earlier Data Encryption Standard (DES). AES' strengths include its security strength, hardware support on current processors, and widespread adoption into standards. The stream cipher ChaCha20, developed by Daniel Bernstein, has gained popularity especially in applications where there is no hardware AES support, like mobile or embedded systems. Yet neither of these ciphers resolves the problem that drives the majority of blockchain cryptography: ensuring trust among parties who were previously unacquainted and who might be motivated at every turn to lie to each other. While symmetric ciphers offer protection when there is already an established key, they necessarily rely on the assumption that this coordination already happened, something that permissionless and trust-minimized networks cannot assume by design. It is not a limitation of AES or ChaCha20, but merely shows the extent of how symmetric cryptography can solve problems,

and why asymmetric ciphers are so essential for blockchains. Picking an algorithm is only half of what the design entails; applying it correctly to any data beyond one block is equally important, and that is the job of a mode of operation. The Electronic Codebook (ECB) mode operates on individual blocks and is now deemed insecure in almost all contexts, because repeated blocks of plaintext result in repeated blocks of ciphertext. In Cipher Block Chaining (CBC) mode, each block is encrypted based on the previous block using XOR, which eliminates the above-mentioned vulnerability, but requires proper use of an initialization vector and lacks built-in integrity. Even more advanced is the Galois/Counter Mode (GCM), which not only offers counter-based encryption, but also adds an authentication tag, thus becoming a form of Authenticated Encryption with Associated Data (AEAD). It is the applications of these cryptographic modes that show themselves in the blockchain technology not in the ledger, where the transactional data is usually not encrypted to be publicly verified, but in the communication channels between nodes. The peer-to-peer communication protocols used in blockchain technology often use either TLS or customized secure channel protocols, in which case they utilize either AES-GCM or ChaCha20-Poly1305 ciphers for encryption of the sessions after establishing a shared session key through an asymmetric key exchange protocol. This combination of asymmetric and symmetric cryptography is a common approach in secure systems and will be considered in detail below.

3.2.2. RSA, Elliptic Curve Cryptography, and Key Management

Symmetric encryption provides an efficient solution to bulk encryption but assumes that both parties have an existing secret key, which is a problem that public-key, or asymmetric cryptography, seeks to solve. Public-key cryptography uses keys which are mathematically related: a public key which can be openly distributed and a private key which has to be kept secret, where any operation done, the use of this key is limited strictly to verified or undone using the other key. The RSA acronym is derived from the surnames of its creators: Rivest, Shamir, and Adleman and refers to one of the early public-key cryptographic techniques, based on integer factorization mentioned in section 3.1.2, where the minimum key size should be at least 2048 or 3072 bits. As compared to its competitors, ECC requires far fewer bits to guarantee the same level of security; a 256-bit ECC key is said to provide the same level of security that a 3072-bit RSA key does since the elliptic curve discrete logarithm problem is thought to be more difficult to solve relative to key length than the problem of integer factorization. This has made ECC the de facto standard when it comes to cryptography within blockchain networks, due to the savings in terms of storage and bandwidth usage from shorter keys and signatures.

The choice of curve is not just a matter of math, but one that has implementation implications that have mattered in the real world. The reason why secp256k1 was chosen for Bitcoin was, in part, because it was, at the time of selection, a less analyzed curve than the NIST recommended curves that had come into question after it came to light that there were some problems with the random number generator for these curves that had been introduced by the NIST. While it might not have been true in this case, the general point here is that the source and openness of the parameter choices for the curve are just as important as their mathematical strength. Having an appropriate algorithm does not equate to handling it securely, and decentralized networks operate key generation processes in a different way compared to centralized networks. The users usually generate keys independently without depending on a third-party system, unlike

centralized PKIs that depend on certificate authorities that connect public keys with authenticated identities. This is how the self-sovereignty of blockchains works; the users' addresses are generated from their public keys, and having the private key is the only way of proving ownership over the related assets. Key exchange techniques are used to establish a shared secret in the absence of any previous channel; examples include Diffie-Hellman and the elliptical curve Diffie-Hellman. Self-sovereignty is not free, as no method for recovering access to a lost key (similar to password resets) exists, meaning that once a private key is lost, all the assets or identity linked to that private key are also lost forever. This one detail alone has fueled most studies related to HD wallets, multisig solutions, and social recovery, as each of these techniques attempts to mitigate the risk posed by a single private key.

There is a price to be paid for all of these options: Larger secret keys and/or more conservative parameters provide a higher level of resistance to cryptanalysis but require a greater number of computational resources (to generate, sign and verify) than smaller keys or less conservative parameters. This is particularly important in blockchain environments where full node verification of many signatures may be necessary and where resource-constrained environments such as IoT devices operate as part of a blockchain/AI ecosystem. The trade-off between resistance to cryptanalysis and computational resource requirements is the basis for the efficient signature schemes and aggregation techniques outlined in Section 3.3.

3.3. HASH FUNCTIONS, DIGITAL SIGNATURES, AND IDENTITY

3.3.1. Hash Function Properties and Merkle Trees

Encryption protects confidentiality; ensuring the integrity of the data remains intact tampered with is a separate problem, and it is the one cryptographic hash functions are built to solve. A hash function maps an input of arbitrary length to a fixed-size digest, and several properties determine whether that mapping is secure. Pre-image resistance requires that, given a digest, finding any input that produces it should be computationally infeasible. Second pre-image resistance tightens this further: given a specific input and its digest, finding a different input with the same digest should likewise be infeasible. Collision resistance, the strongest of the three, requires that finding any two distinct inputs sharing a digest be infeasible a property bounded in theory by the birthday paradox, but one that well-designed hash functions approach as closely as their output length permits.

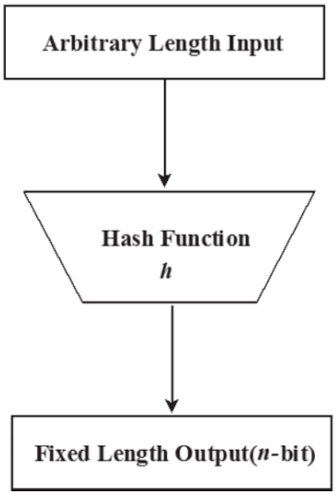


Fig 4: Block diagram of Hash Function

A related property, the avalanche effect, requires that a minimal change to the input even a single bit produce a drastically different, seemingly unrelated output. The reason hash functions are useful for ensuring integrity is that any changes made to a piece of data (regardless of how small the change is) would result in such a significant difference between the original hash and the new hash that it can be detected. There are several hashing algorithms in existence that fit these descriptions, but the most widely used is the SHA-2 hashing algorithm (or its variants, which include SHA-256 and SHA-512), which was adopted as a standard by NIST. Bitcoin, for example, uses SHA-256 as part of its proof-of-work system and block hashing scheme. There is also SHA-3, which was developed much later and has an entirely different internal structure (the sponge structure) than SHA-2, which uses the Merkle-Damgård structure. One reason for the introduction of SHA-3 was to protect against any future discoveries regarding the cryptographic weaknesses of SHA-2; thus far, no such discovery has occurred and so SHA-2 has proven to be secure enough. Adding to the complexity of the overall hashing landscape is Ethereum's use of Keccak-256 (which is closely related to but separate from the formalized SHA-3 hashing algorithm) because they both use different padding schemes. This illustrates the fact that hashing technologies continue to evolve and that there are still differences among them based on how each platform implements them within their own systems (i.e., hashing technologies are not yet fully standardized).

When combined into a structure, hash functions can be very effective, and there is no more important structure for blockchain designs than invented by Ralph Merkle, this cryptographic data structure functions as a binary tree. Within this framework, individual leaf nodes contain the hashes of distinct data blocks, while every internal node stores the combined hash of its direct descendants. This arrangement produces a single 'root hash' that can serve as a compact summary of the integrity of an entire dataset. Each block of a blockchain will typically arrange the transactions contained within it into a Merkle Tree format; as a result, all transactions contained in the block can be referred to by a single Merkle 'root' contained in the block header. The benefit of this structure is known as a Merkle proof; therefore, a party seeking to verify that a particular transaction is a part of a specific block of the blockchain can confirm the validity of the transaction by verifying a logarithmic number of hash values instead of needing to verify every transaction included in the block - saving verification time for lightweight or resource-constrained clients that do not have the storage capacity for the entire blockchain.

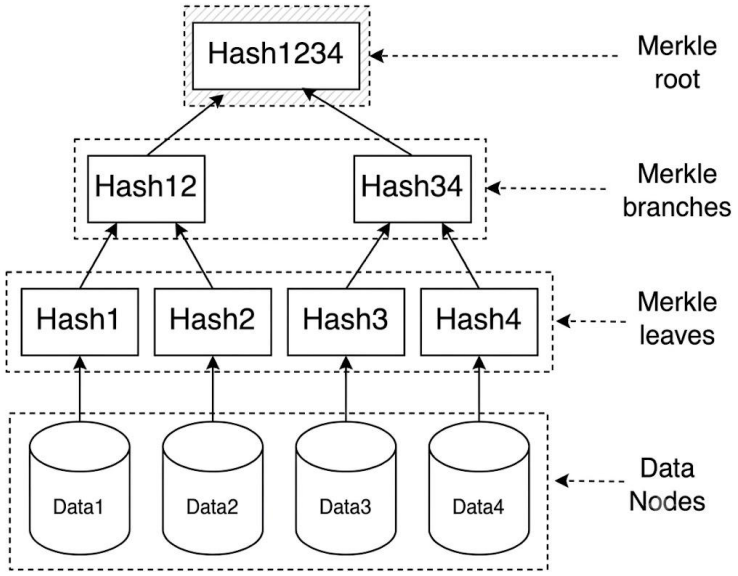


Fig 5: Block diagram of Merkle Trees

One practical example of why this efficiency is important is in verifying that a block of one million transactions, organized as leaf nodes of a Merkle Tree, includes a particular transaction. To verify that a transaction is part of the block, you only need to check approximately 20 hash values (specifically, the base 2 logarithm of 1 million) instead of checking every single transaction in the block (1 million transactions). Each of these 20 hash values corresponds to the hash of the tree's sibling node, from the leaf (transaction) node to the tree's root. By re-calculating and comparing the chain of hash values from the leaf to the root against the Merkle Root that is located in the block header, you are able to confirm that the transaction is included in the block as surely as you would if you had verified 1 million transactions. It is the logarithmic nature of this scaling that allows for light clients (like mobile and embedded devices) to participate in the blockchain verification process without having to bear the costs of storing and transferring all of the data necessary to maintain a complete copy of the blockchain (as would be done with an archival node).

3.3.2. Digital Signature Schemes

Digital signatures allow a user with a private key to create a message that can be verified by someone with the associated public key due to the way that they utilize the cryptographic techniques discussed in Section 3.2.1 (i.e. Hash functions provide assurance that records have not been modified, while digital signatures also establish the source of creation of the data). ECDSA (Elliptic Curve Digital Signature Algorithm), an asymmetric key mechanism based on hardness assumptions, is the most utilized signature mechanism on blockchain networks (for example Bitcoin and Ethereum use ECDSA over the secp256k1 curve).

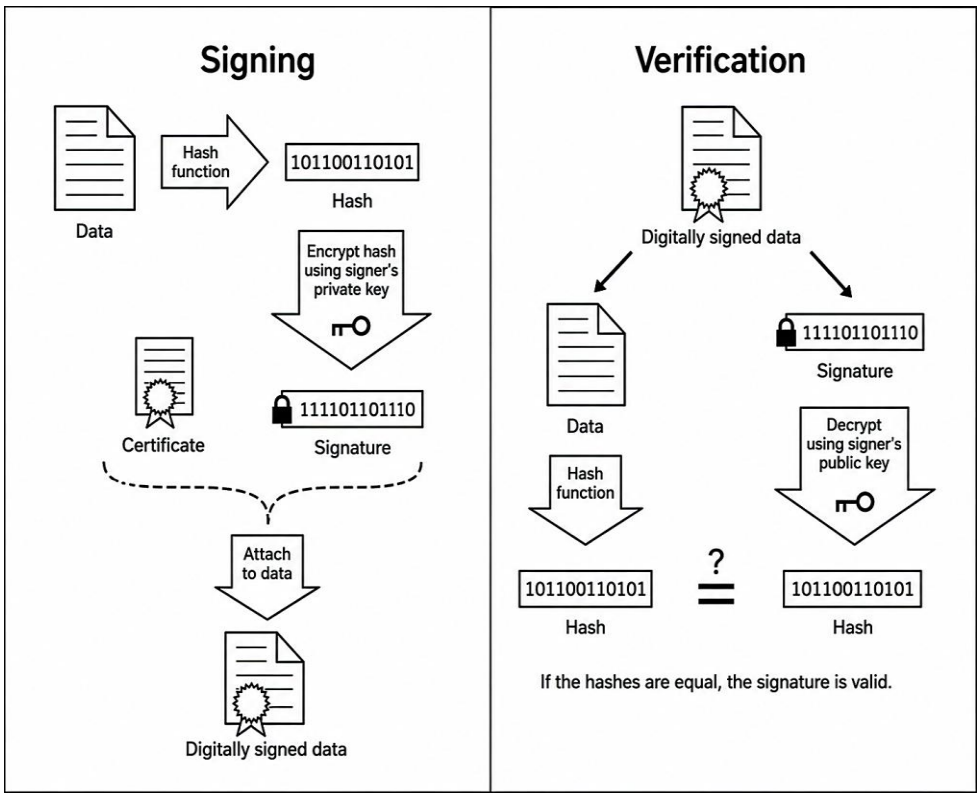


Fig 6: Block diagram of Digital Signature

As a newer option, EdDSA (Edwards-curve Digital Signature Algorithm), has begun to be utilized by various blockchain projects and applications due to its higher performance, simpler and less prone to error implementation requirements (for example, EdDSA supports deterministic signing eliminating ECDSA's risk from weak or reused random nonces), as well as resistance to various side channel attacks. The differences between the types of signature verification provided by the CDSA and ECDSA illustrate the distinction between cryptology's strength and the security of its implementation. Due to their respective implementations, both signature types, CDSA and ECDSA, can resist attacks on their underlying strength when properly executed; however, the existence of a poorly designed random number generator would lead to the establishment of a vulnerability in the ECDSA signature creation process as it relates to the reliability and uniqueness of the generated nonces and their subsequent generation. For example, over time, an attacker exploiting the same RNG-generated nonce would have enough information to derive the private key belonging to the original signer through the generation of multiple ECDSA signatures (as opposed to generating multiple signatures using the same ECDSA nonce). As a result of this, many compromised key keys occur as the result of mistakes made by implementers of the ECDSA algorithm. By using the EdDSA algorithm, which does not use the RNG to generate a nonce for each signature, implementers of EdDSA do not face the same potentiality for key compromise as they would with the ECDSA algorithm. As a result of this, implementing EdDSA will not result in the implementation of the algorithm presenting any substantial increase in the likelihood that a cryptographic key would be compromised. In general, the overall security of a crypto-system following deployment is not dependent solely upon the resistance of the crypto-system to an attack (based upon the difficulty of solving the underlying mathematical problem); rather, it is also dependent upon the number and severity of mistakes made by the implementer.

Risk is centralized in a single point using a single key. To alleviate that risk, there are two different types of schemes available. A multi-signature scheme allows you to validate a transaction or message by requiring that it be independently signed by a minimum number of different private keys, an example of which is the M-of-N policy which allows for only M signatures to be valid if there are N total keys. This method has become the industry standard for distributing custodial control of crypto-assets, thus reducing the risk associated with compromising or losing any single key. Another type of scheme which accomplishes the same goal as a multi-signature scheme is a threshold signature scheme (TSS). Whereas multi-signature schemes produce and verify multiple signatures, TSS combines a designated number of participants into a group to create one signature that is indistinguishable from an individual signature examined by an outside party. This method has two major advantages: the group has one signature, which takes up less space, and the multi-party structure used to create the signature is not public information, whereas when you use a multi-signature method to provide proof of an asset you can see it on the blockchain.

3.3.3. Identity in Permissioned and Permissionless Systems

Signature-based identity depends on how institutions structure their governance. The governance structures of permissionless networks (like Bitcoin and Ethereum) permit people to create keys without any authorization and to participate based solely on cryptographic proof of their possession of the key. Therefore, they require no identification in terms of a verified identity in the real world. In contrast, permissioned networks utilize a public key infrastructure whereby there has been a formal identification process prior to a person being authorized to create a key and participate by a governing body or consortium. Such arrangements are common in enterprise or consortium implementations where there are government (or other) regulations or the need for an accounting structure that supports a workable method of auditability. These two forms of signature-based identity have direct ramifications on AI and blockchain workflows that incorporate AI-based decisions; the guaranteed identity of permissioned networks alleviates the complexities associated with tracking model/decision accountability, while the pseudonymity associated with permissionless networks provides a lesser degree of accountability with greater confidentiality. Neither approach provides a greater degree of reliability, so the choice between them must be based on the intended use of the implementation (i.e., whether their goal is to allow open access to the public or to provide a system of institutional accountability).

3.4. ADVANCED CRYPTOGRAPHIC CONSTRUCTS FOR SECURE COMPUTATION

3.4.1. Zero-Knowledge Proofs

The primitive constructs previously described demonstrate protection of data and verification of identity; the next constructs extend beyond these features to enable proof of computation, in addition to preserving the characteristics of trust, to conceal computations from being directly observed and to distribute computations without compromising trust. The first such construct is the 'zero-knowledge proof.' through Zero-Knowledge Proofs, a proving party can convince a verifying party of a claim's accuracy without disclosing any supplemental data beyond the assertion's truth ZKPs must be proven as providing 'completeness' (i.e., a true statement can

be proven) and 'soundness' (i.e., a false statement cannot be proven except with minimal probability) and also that the verifier cannot learn anything beyond the fact of validity of the given statement. The most predominant current implementations of ZKPs include two general categories: zk-SNARKs (i.e., zero-knowledge succinct non-interactive arguments of knowledge) which provide succinct (i.e., small), fast to verify and non-interactive proofs (i.e., once an initial set-up phase has been completed no further communications between the prover and the verifier are required) of any type of computation due to their succinctness but require a trusted ceremony to establish the parameters for the proof generation process, thus severely limiting the ability to provide for an untrusted environment in which to apply the zk-SNARK framework; and zk-STARKs (i.e., zero-knowledge scalable transparent arguments of knowledge) which provide for the elimination of the trusted ceremony requirement thus providing for the ability to establish an untrusted proof generation environment while still enabling a verifier to produce proofs that can be utilized to establish trust between parties.

Instead of using only trusted setups, they can be based on collision-resistant hash functions, leading to proof sizes larger than zk-SNARKs. Many Blockchain Systems already use both types of proofs; ZKP is used for the privacy protection of coins such as Zcash, and more and more to provide layer 2 Scaling Solutions called ZK-Rollups that use succinct Proofs (ZKP) of validity for large off-chain transactions without requiring those transactions to be executed again on-chain. The scaling implications of ZKP are worth elaborating on, as they provide a powerful way to show how zero-knowledge proof technology will change the balance between trust and efficiency, a key design characteristic of all Blockchain systems. For instance, with large off-chain transactions, all nodes would typically need to re-execute all of those off-chain transactions in order to validate their authenticity (a linear process and quickly becomes a limiting factor on the size of the network), unless succinct proofs are used. That way, no matter how many transactions are included in an off-chain group, there is one or more succinct proofs that would, in total, fit inside those same transactions, and thus the amount of time it takes to validate one or more succinct proofs would be approximately the same, regardless of how many transactions exist in an off-chain group.

3.4.2. Homomorphic Encryption

Where ZKPs prove a computation happened correctly, homomorphic encryption lets the computation itself run on encrypted data, producing an encrypted result that, once decrypted, matches what plaintext computation would have yielded. Partially homomorphic encryption (PHE) supports only a single operation typically addition or multiplication applied an unlimited number of times to ciphertexts; the Paillier cryptosystem, supporting homomorphic addition, is the most frequently cited example.

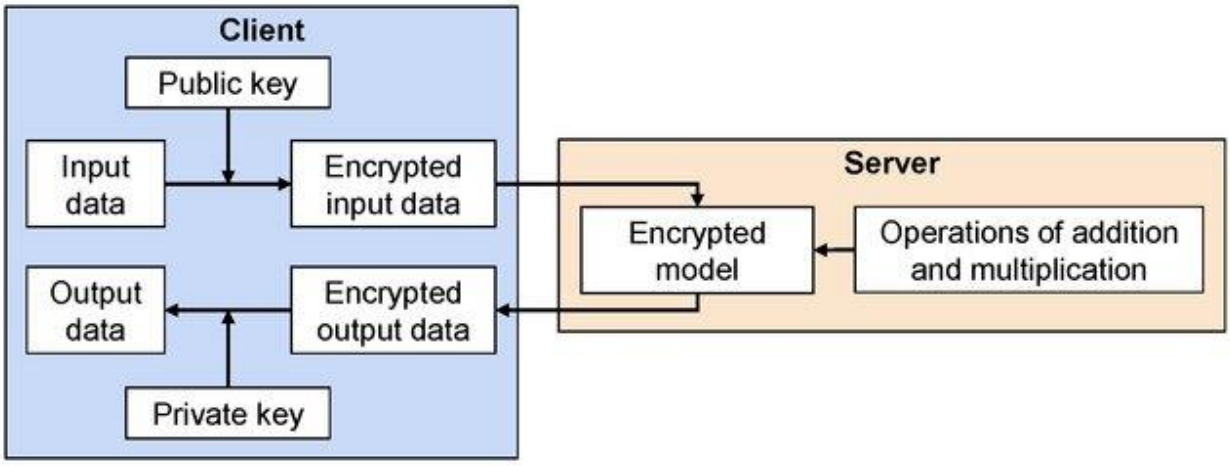


Fig 6: Block diagram of FHE

Fully homomorphic encryption (FHE) goes further still, supporting arbitrary computation over encrypted data. First realized in a practical construction by Craig Gentry, FHE combines a "somewhat homomorphic" scheme with a bootstrapping procedure that periodically refreshes ciphertext noise. Its generality makes it attractive for secure outsourced computation including AI inference over encrypted data, taken up in Section 3.5 but the computational overhead remains steep, often several orders of magnitude slower than plaintext computation, which keeps research active on more efficient schemes such as CKKS, BGV, and BFV. Some intuition for why this overhead is so substantial helps explain why FHE remains a research frontier rather than a deployed default. Plaintext arithmetic operates on numbers directly; homomorphic arithmetic instead operates on ciphertexts that encode noisy approximations of those numbers, and each operation performed on a ciphertext increases the accumulated noise. Left unchecked, this noise eventually overwhelms the signal and renders the ciphertext, which is exactly the problem Gentry's bootstrapping procedure exists to solve, at the cost of periodically performing an expensive noise-reduction operation partway through a computation. The practical upshot is that an FHE-based computation does not simply run slower than its plaintext counterpart by a fixed multiplicative factor; the overhead compounds with the depth and complexity of the computation being performed, which is precisely why FHE has so far found more traction in computations with shallow, well-bounded structure than in the deep, iterative computations characteristic of training a machine learning model from scratch.

3.4.3. Secure Multi-Party Computation and Secret Sharing

Homomorphic encryption keeps a single party's data private from everyone else; secure multi-party computation (SMPC) addresses the related but distinct case in which several mutually distrusting parties want to jointly compute a function over all of their private inputs, without any party learning more than what the final output implies about the others' data. The establishment of Yao's garbled circuit construction framework for secure evaluation of functions (two-party setting) and the GMW protocol's framework for multi-party setting, is fundamental to understanding that all computable functions can be securely evaluated, while also demonstrating the differences in practical efficiency based on both the function's structure and the protocol used to securely evaluate that function.

In Section 2 of this article, there are several secure multi-party computation (SMPC) protocols which leverage a more elementary concept of secret sharing, and therefore create shared private data among participants in such a way that they can combine their portions and reconstruct the shared secret while less than the minimum number of participants will have any knowledge of the secret. The most famous example of a secret sharing construction is Shamir's secret sharing scheme that uses polynomial interpolation over a finite field to perform the sharing. Shamir's secret sharing is implemented as a (t,n) scheme, and therefore any group of t out of the n total number of participants may combine their shares in order to reconstruct the secret. VSS is a new type of secret sharing that allows a participant to confirm whether the shares they received were distributed according to the rules of the secret being shared and allows them to do so without having any confidence in the dealer who provided them with the shares. In addition, since there is no single party in the network that has enough trust to give any participant a guarantee of the validity of their share without the use of a trusted third-party system, the ability of VSS to allow for verification will make it more appropriate for constructing threshold signatures. The use of Distributed Key Generation (DKG) protocols and related schemes across various areas of blockchain consensus and governance has greatly opened up new possibilities. Specifically, these protocols can allow organizations to securely share cryptographic keys with custodians while limiting their access to those keys depending on various conditions.

To illustrate how useful the threshold property is, I will present a simple numerical example. Let’s say an organization wants to ensure that there are five custodians (C1-C5) and that any three of them can reconstruct the cryptographic key while no two of them can recover any information about the key. The organization can accomplish this goal by implementing a (3,5) Shamir scheme as described below.

The organization acts as the dealer and encodes the key as the constant term of a randomly chosen quadratic polynomial (a polynomial of degree 2). The organization selects five unique points, uses these points to evaluate its secret polynomial at each of these five unique x values, and distributes these evaluation results to the custodians. The three custodians who have been chosen to reconstruct the secret can interpolate the polynomial to recover the value of the constant term. However, since only two custodians can work together, they cannot recover the constant term. This guarantee that two custodians will not learn anything about the constant term if they are only able to access two evaluation results by cooperating is not just a practical issue for an attacker in possession of two pieces of the evaluation result; it is also a theoretical impossibility since each pair of points is consistent with an infinite number of quadratic polynomials, each corresponding to a different value for the constant term. The same type of analysis can be scaled up as needed to explain the distributed custody models increasingly being used to protect high-value digital assets on blockchain systems and the cryptographic keys that underpin the consensus mechanism and overall governance models used to manage most blockchain applications as described above.

3.5. CRYPTOGRAPHY FOR AI–BLOCKCHAIN CONVERGENCE

3.5.1. Privacy-Preserving Computation for Federated Learning

Federated learning is a form of shared model training that enables many parties to collaborate to train a common model without bringing raw training data into a central location. Instead, model updates are sent to an aggregator for aggregation into a global model. Although there is some degree of privacy benefit associated with federated learning as compared to centralized training, there is also a risk of inference attacks being possible on individuals whose underlying training data were used to produce the model updates sent to the aggregator; this is why secure aggregation protocols utilize secret sharing or secure multi-party computation (SMPC) techniques, as described above, to allow the aggregator to compute a sum or averaged value of the total number of aggregated model update contributions but not be able to view any individual contributions in their original clear-text form. In addition to secure aggregation, differential privacy is often implemented as an additional layer protecting the overall system by adding noise to the model update, which helps ensure that formal guarantees are maintained to limit the ability of individuals to conduct inference attacks; differential privacy is complementary to the cryptographic protections provided by the previously mentioned secret sharing and SMPC techniques. Federated learning using blockchain technology frequently implements incentive mechanisms as a means of coordinating participant cooperation despite the complete absence of any trust relationship between any of the parties involved, effectively transforming a privacy-focused technique into a viable economic platform.

3.5.2. Verifiable Computation for AI Model Integrity

While securing the actual training of an AI model presents unique challenges, verifying the output from that model after it has been trained represents a separate (but growing) challenge especially given the increasing use cases for AI models, such as making significant decisions, including but not limited to credit scoring, medical diagnosis, and controlling autonomous systems. Once model execution has transitioned from on-chain to off-chain for efficiency purposes, an established trust gap exists with the ability to verify whether a specific output is actually generated from a specific model, and then, if so, whether that model actually processed a specific input. In addressing this trust gap, zeros-knowledge machine learning (zkML) proposed novel research into leveraging zk-SNARK and zk-STARK, as previously discussed, to generate succinct proofs that a specific model produces a specific output from a specific input, without requiring the disclosure of model parameters or input data. The benefits associated with zkML extends beyond providing model provenance and auditing the decisions made by AI in regulated areas, but also can be used as a foundation for incorporating trust into off-chain inferences into blockchain solutions so that the outputs produced by the AI can be used by the blockchain, while also preserving the level of trust associated with blockchain technology.

3.5.3. Cryptographic Challenges Unique to AI Workloads

While these techniques hold great potential, some of the challenges surfaced when using them for AI workloads differ not only in degree but sometimes also in kind from those for the original blockchain architectures. One of the main challenges is the scale of the data and parameters associated with AI training, which are generally in the gigabytes-to-terabytes range, compared to the much smaller workloads on which FHE and SMPC were designed and tested. This will create substantial scalability issues when the new techniques are used at AI scale. In addition, significantly increased latencies will make this issue even more complex. AI applications that require real-time inference often cannot handle the heavy computational load imposed by the

strong security features of FHE and therefore reflect a trade-off between data privacy and responsiveness, unlike traditional blockchains that account for latency when settling large volumes of transactions. A third, distinct concern is model confidentiality: where blockchain cryptography has traditionally protected data and identities, AI deployments increasingly need to protect the model itself, its architecture, and trained parameters as valuable intellectual property. The data-confidentiality-focused primitives discussed earlier in this chapter do not fully address this.

3.6. POST-QUANTUM CONSIDERATIONS, OPEN CHALLENGES, AND CONCLUSION

3.6.1. The Quantum Threat and Standardized Responses

Every scheme discussed up to this point assumes a classical adversary, and that assumption has a known expiration date. RSA, ECDSA, and related public-key schemes rest on hardness assumptions of integer factorization and discrete logarithm problems that classical computers cannot feasibly break but that a sufficiently large, fault-tolerant quantum computer running Shor's algorithm could. Grover's algorithm poses a milder threat, offering only a quadratic speedup for brute-force search, including symmetric key search; this effectively halves AES's practical security margin against a quantum adversary, a concern addressed simply by doubling key lengths, in sharp contrast to the more fundamental threat Shor's algorithm poses to asymmetric schemes. No quantum computer capable of running Shor's algorithm against currently deployed key sizes exists as of this writing, but blockchain-recorded data tends to outlive short-term predictions, and the timeline to practical quantum computing remains genuinely uncertain, reason enough to take proactive standardization seriously well ahead of need. NIST's response was a multi-year post-quantum cryptography standardization process, which selected several candidate schemes believed to resist quantum cryptanalysis and finalized them as Federal Information Processing Standards in August 2024. Lattice-based schemes lead the pack: ML-KEM (FIPS 203, standardizing the original CRYSTALS-Kyber submission) and ML-DSA (FIPS 204, standardizing the original CRYSTALS-Dilithium submission) both rely on the Learning-With Errors hardness assumption introduced in Section 3.1.2, and together represent the most prominent family of standardized post-quantum schemes thanks to a favorable balance of security, performance, and key size.

Hash-signature systems like SLH-DSA (FIPS 205, which standardizes the original SPHINCS+ submission) are built upon conservative principles by deriving their security directly from the hash function's properties outlined in Section 3.3.1, providing very strong assurances backed by well-researched primitives while creating larger signatures than those made with lattice-based systems to obtain this level of reliability. Code-based systems – typified by the well-researched classic McEliece system – offer another dimension of cryptographic diversity through the use of the hardness of decoding random linear codes to decrease the chances of one mathematical breakthrough compromising multiple standardized systems simultaneously. The standardization of new schemes to facilitate migration for live blockchains: migrating live blockchains to new schemes is the more difficult task to accomplish. The majority of platforms derive their account addresses from the public key; thus, the ability to successfully migrate requires addressing both the signing components for future inputs as well as the security of the

historical public keys; these still exist and will be vulnerable to rekeying attacks once quantum computing capabilities mature (this is referred to as "harvest now/decrypt later"). There are hybrid schemes designed to create possibilities for a successful transition by using a composite algorithm constructed from combining a classical algorithm with a post-quantum algorithm to generate account addresses with the intention that either algorithm remains unbroken, thus providing assurance of security for near-term weak implementations of newly developed post-quantum schemes in addition to security from quantum computing after a longer period of time through the current classical algorithms.

3.6.2. Open Implementation Challenges

Based on the previous analysis of constructs addressed in this chapter, three common issues are apparent:

1. Performance - Fully Homomorphic Encryption (FHE) and general-purpose SMPC (or secure multi-party computation) are both orders of magnitude greater in cost (computationally) than their plaintext counterparts. The result of this discrepancy has been a persisting trade-off between the strength of the privacy/integrity guarantees that a system can provide and its actual ability to provide those guarantees in terms of practical throughput or latency of that system, particularly as these constructs begin to operate at the types and amounts of data and model volume that are commonplace in AI related workloads.

2. Key Management - As decentralized networks move away from individual human participants toward many additional participants (such as IoT devices and autonomous AI agents), key management will continue to become more complex on two fronts.

3. Institutional Factors: The third gap comes from institutional factors, rather than from technical factors; there has been significant progress toward establishing standards for the three primary primitive components of blockchain and artificial intelligence convergence (i.e., blockchain protocols, artificial intelligence algorithms, and zero-knowledge proof systems). The primary primitives that compose the actual blockchain and AI systems are not yet settled; for example, A, for instance, is a zero-knowledge proof system. They have been mathematically proven to be valid and reliable; however, many implementations and providers of these systems operate as stand-alone proof systems, making it very challenging to compose these components from multiple blockchain and artificial intelligence systems. Additionally, zkML does not use any existing standards for expressing verifiable computation claims in either zero-knowledge proof systems or zkML. Therefore, the above three gaps present opportunities for future scholarly research by the information security, cryptography, and artificial intelligence research communities.

3.6.3. Conclusion

This chapter has traced the cryptographic foundations of secure computation in blockchain and AI-converged systems, from the four core security properties confidentiality, integrity, authenticity, and non-repudiation through the symmetric and asymmetric primitives, hash functions, and digital signatures that implement them, and on to the more advanced constructs: zero-knowledge proofs, homomorphic encryption, secure multi-party computation, and secret sharing that make privacy-preserving and verifiable computation possible in decentralized

settings. Particular attention was given to the distinctive demands AI workloads place on this cryptographic layer, including the scale, latency, and model-confidentiality challenges that set AI-blockchain convergence apart from blockchain's original financial-transaction use case. None of the primitives surveyed here function in isolation; each gains its practical force from combination with the others: hashes feeding signatures, signatures feeding multi-party protocols, multi-party protocols feeding the privacy-preserving learning systems that make AI-blockchain convergence trustworthy rather than merely possible. As blockchain and AI systems continue to converge, the cryptographic layer surveyed in this chapter remains the bedrock on which trust, in the absence of central authority, is mathematically constructed.

CHAPTER 4

CONSENSUS MECHANISMS AND ALGORITHMIC TRUST

Contributed by: Tridip Mazumder, Rishov Chakraborty, Diwakar Pandit, Rohit Chawhan

4.1 INTRODUCTION

Trust, in classical social theory, is a disposition toward vulnerability grounded in reasonable expectations of another party's behaviour. Where individuals are anonymous, geographically dispersed, and not beholden to any central authorities, there can be no such reliance on interpersonal relationships, institutional affiliation, or legal responsibility. The main thesis of this chapter is that consensus algorithms are far from just technical issues but rather represent computation itself as an expression of trust. Satoshi Nakamoto's 2008 white paper established that a cryptographically secured chain of verifiable computational puzzles could render dishonest behaviour economically irrational, substituting algorithmic rule for social trust. This is really an innovative philosophical stance: Code, instead of humans or organizations, may be what makes agreement possible between strangers. The rise of consensus mechanisms after 2009, from Proof of Work to Proof of Stake to Byzantine Fault Tolerant families and beyond, represents a sustained attempt to realise, refine, and extend this proposition across a growing diversity of deployment contexts.

This chapter provides a systematic treatment of that proliferation. It opens with the theoretical foundations constraining the design space — the Byzantine Generals Problem, the CAP theorem, and the blockchain trilemma — then examines the principal protocol families in graduated depth before conducting a comparative analysis and developing an original conceptual framework of algorithmic trust. Throughout, the analysis attends to both formal properties and practical deployment realities, grounding theoretical claims in quantitative benchmarks and operational case studies.

4.2. THEORETICAL FOUNDATIONS OF DISTRIBUTED CONSENSUS

Three theoretical constructs carry the greatest analytical weight for understanding consensus protocol design: the Byzantine Generals Problem, the CAP theorem, and the blockchain trilemma. Together, these frameworks define what is possible, what is impossible, and what must be traded away in any distributed consensus system.

4.2.1 The Byzantine Generals Problem

In their landmark 1982 paper, Lamport, Shostak, and Pease introduced The Byzantine Generals Problem,In the famous 1982 paper by Lamport, Shostak and Pease, the concept of the Byzantine Generals Problem is illustrated with a story about the military world. In the tale, the generals communicate with each other using messages from messengers, some of which may be traitors spreading false information. Thus, the problem lies in finding a protocol that ensures agreement among all loyal generals regardless of the number of adversaries.

The mathematical answer is formally exact. A distributed system of n nodes can tolerate at most f Byzantine — arbitrarily malicious or faulty — nodes if and only if $n \geq 3f + 1$. Communication complexity is $C(n) = O(n^2)$ for classical BFT protocols, meaning every additional validator multiplies total message traffic. This quadratic scaling is the central engineering constraint that subsequent BFT work, culminating in HotStuff's O(n) linear complexity, has sought to reduce. Castro and Liskov's Practical Byzantine Fault Tolerance (PBFT, 1999) was the first deployable BFT protocol and remains the direct ancestor of consensus mechanisms used in enterprise blockchain platforms such as Hyperledger Fabric today.

4.2.2 The CAP Theorem and Blockchain Constraints

The CAP Theorem and Blockchain Constraints Brewer's CAP theorem, formally proved by Gilbert and Lynch in 2002, asserts that no distributed data system can simultaneously guarantee Consistency (every read returns the most recent write), Availability (every request receives a response even under failures), and Partition Tolerance (the system continues when the network splits). For any blockchain deployed across an open and unreliable network, partition tolerance is non-negotiable; the operative design trade-off is therefore between consistency and availability. Permissioned BFT platforms such as Hyperledger Fabric prioritise consistency — halting block production rather than risking divergent ledger states. Permissionless chains such as Bitcoin prioritise availability, accepting temporary divergence. A new CAP-like framework specific to autonomous consensus systems proposes that at most two of three properties — Consensus achievement, Autonomy, and entropic Performance — can be simultaneously optimised. This formalisation provides a mathematical account of the trilemma trade-offs that practitioners observe empirically and is treated here as a structuring heuristic grounded in cross-protocol observation.

4.2.3 The Blockchain Trilemma

The blockchain trilemma, widely attributed to Vitalik Buterin, holds that no blockchain system can simultaneously maximise Decentralisation, Security, and Scalability. This chapter treats the trilemma not merely as an empirical observation but as a genuine structural feature of the consensus design space, in line with the formal constraints discussed above. The consensus protocols studied in Sections 4.2 to 4.4 each exemplify one resolution approach that emphasizes some features at the expense of others, and Section 4.6 discusses this in a comparative manner.

4.3. PROOF OF WORK: THE GENESIS OF ALGORITHMIC TRUST

4.3.1 Mechanism and Operation

Proof of Work requires each mining node to find a nonce such that the SHA-256 hash of the block header falls below a dynamically adjusted target threshold. Because hash functions are one-directional, the only effective strategy is brute-force iteration across trillions of candidate values per second. A difficulty adjustment mechanism recalibrates the target approximately every 2,016 blocks (roughly two weeks) so that the mean time to find a valid block remains approximately ten minutes regardless of total hashing power.

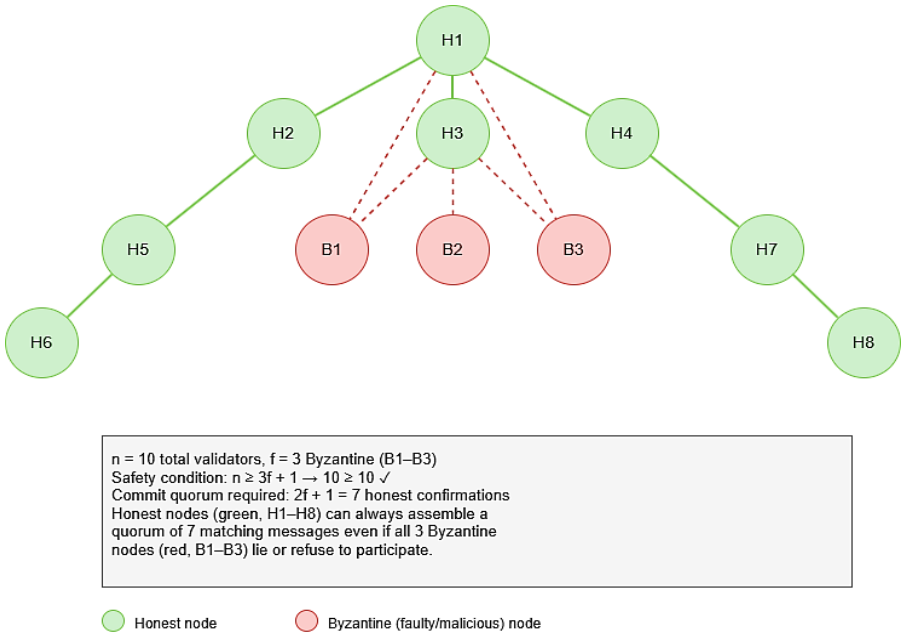


Figure-1 Byzantine Quorum

This self-regulating property reveals a fundamental constraint: adding more computational resources increases security but not throughput. Bitcoin's throughput is architecturally capped at approximately 7 TPS — a figure derived from the combination of a fixed block size and a fixed 10-minute block interval. Ethereum, before the Merge, operated at approximately 12–15 TPS under PoW. This ceiling is structural rather than incidental, rooted in the protocol's commitment to maximising decentralisation and security.

4.3.2 Security Properties and Attack Vectors

PoW's foundational security assumption is that the network remains safe as long as honest miners collectively control more than 50% of total hashing power. An attacker achieving majority control — the 51% attack — can mine a concealed fork and release it to reverse confirmed transactions and execute double-spend fraud. For Bitcoin, with estimated annualised hashing power exceeding 600 exahashes per second in early 2025, the capital and energy cost of sustaining such an attack vastly exceeds any plausible gains. The same deterrence is substantially weaker for smaller altcoin networks: Ethereum Classic suffered multiple 51% attacks in 2019 and 2020, including an August 2020 incident reversing over 4,000 blocks and enabling double-spends worth millions of dollars. The selfish mining attack, described by Eyal and Sirer in 2014, demonstrates that a pool controlling as little as 33% of hashing power can achieve disproportionate block rewards by strategically withholding discovered blocks. Eclipse attacks — manipulating network topology to isolate specific nodes — represent a complementary vulnerability at the network layer rather than the consensus layer.

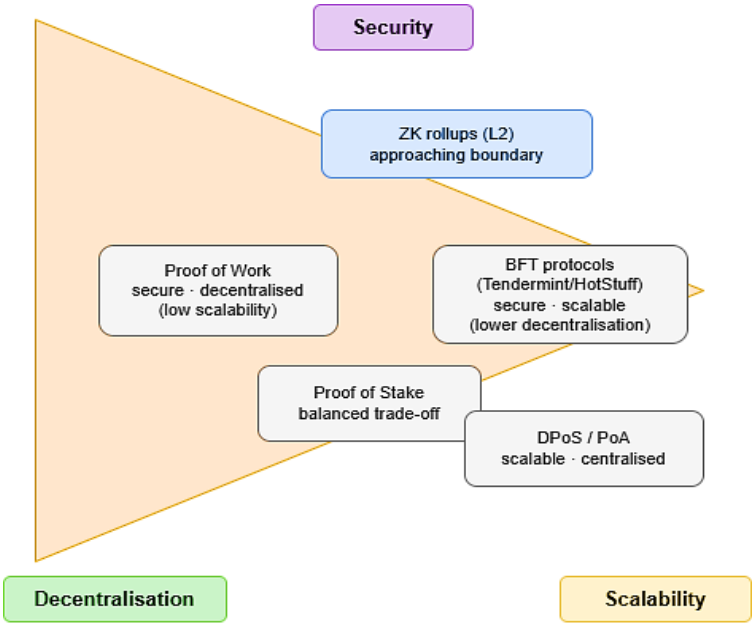


Figure-2 Blockchain Trilemma

4.2.3 Energy Consumption and Environmental Critique

PoW's energy consumption is a feature rather than a flaw: the mechanism through which computational investment is converted into algorithmic trust. Bitcoin's annualised electricity consumption has been estimated at approximately 127 TWh in 2023 — comparable to Norway's annual consumption — with a per-transaction energy footprint roughly one million times greater than a Visa payment. Estimates of the renewable proportion of Bitcoin mining energy range from 25% to over 70%, depending on methodology. The case for PoW's energy expenditure rests on the argument that the security properties purchased — immutability grounded in physical thermodynamic reality — are uniquely valuable and cannot be achieved comparably by any other known mechanism. The case against is equally serious: the throughput produced relative to energy consumed is negligibly small by any utilitarian metric, and the environmental costs are externalities borne by the physical world. This debate illuminates a fundamental tension in algorithmic trust: trustlessness achieved through PoW carries a real-world cost imposed on energy infrastructure.

4.4. PROOF OF STAKE AND ITS PRINCIPAL VARIANTS

The family of Proof of Stake consensus mechanisms emerged as the primary alternative to PoW and became, with Ethereum's landmark transition in September 2022, the dominant consensus paradigm for large-scale public blockchains. The core innovation of PoS is replacing computational work as the Sybil resistance mechanism with economic stake: validators lock up a quantity of the network's native token as collateral, pledging it as a guarantee of honest behaviour.

4.4.1 Classical Proof of Stake

In classical PoS, the probability of selection to propose and validate the next block is proportional to the size of a validator's stake, creating an incentive structure in which greater economic commitment confers greater influence while simultaneously creating greater exposure to penalties for dishonesty. Ethereum's implementation — built on the Casper FFG protocol and the LMD-GHOST fork-choice rule — requires validators to deposit a minimum of 32 ETH as collateral. As of early 2025, approximately 33 million ETH is staked across roughly 900,000 active validators, with theoretical finality latency of approximately 12–15 minutes for Casper FFG checkpoints and single-slot finality proposals in active development targeting a reduction to approximately 12 seconds.

The Nothing-at-Stake problem — in which rational validators might attest to multiple competing forks simultaneously, since doing so costs nothing under a naive PoS design — is addressed through slashing mechanisms that impose severe financial penalties on validators signing contradictory blocks. The Terra/Luna collapse of May 2022 is instructive as a PoS governance failure: despite technically sound BFT properties at the consensus layer, a spiral of algorithmic stablecoin redemptions catastrophically depleted the validator incentive structure, demonstrating that consensus-layer safety does not guarantee protocol-level economic stability.

4.4.2. Delegated Proof of Stake

Delegated Proof of Stake introduces a democratic electoral layer into the PoS framework: token holders vote proportional to their stake to elect a small panel of delegates — twenty-one in the EOS model — authorised to produce blocks on behalf of the broader community. This arrangement provides significant throughput advantages: EOS has demonstrated T > 3,000 TPS with finality latency of approximately 0.5 seconds. However, DPoS's centralisation dynamics are troubling: empirical analyses of EOS governance have documented voting cartel behaviour in which a small coalition of large token holders coordinated vote-trading arrangements, effectively rendering the block-producer election a closed club. DPoS achieves high T and low L while substantially compromising decentralisation — a clear instance of the trilemma.

4.4.3 Leased Proof of Stake

Leased Proof of Stake, implemented on the Waves platform, allows any token holder to lease their stake to a full node without transferring ownership, enabling the leasing node to benefit from a larger effective stake while the lessee retains custody and receives a proportional share of transaction fees. LPoS achieves approximately 1,000 TPS with finality latency of several seconds. Its significance extends beyond performance metrics: the mechanism preserves alignment between economic stake and network security while distributing operational burden to better-resourced nodes without requiring token holders to surrender custody or governance rights. This makes LPoS particularly suitable for resource-constrained IoT deployments.

4.5 BYZANTINE FAULT TOLERANT CONSENSUS PROTOCOLS

BFT protocols assume a known, authenticated set of participants — and in return provide considerably stronger formal guarantees of safety and immediate finality. As established in

Section 4.1.1, BFT safety requires $n \geq 3f + 1$, with communication complexity C(n) varying critically across protocol generations.

s

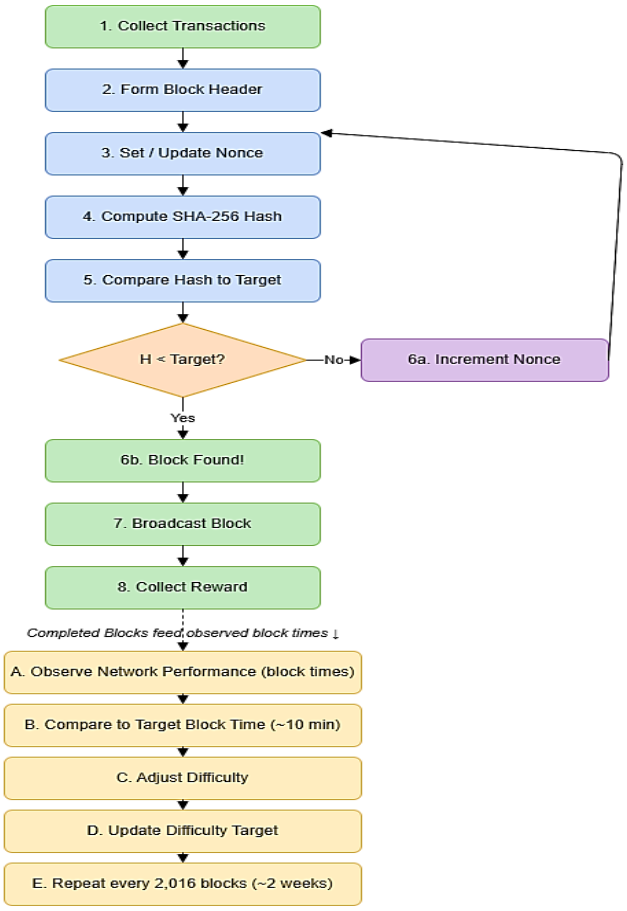


Figure-3 PoW Flowchart

4.5.1 Practical Byzantine Fault Tolerance

PBFT operates through a three-phase protocol: in the pre-prepare phase, the designated primary replica broadcasts a proposed block; in the prepare phase, each accepting replica broadcasts a prepare message to every other replica; and in the commit phase, a replica proceeds only after accumulating $2f + 1$ matching prepare messages, then broadcasts a commit message. A block is committed only when $2f + 1$ commit messages have been received, providing a safety guarantee even in asynchronous networks. PBFT's most important practical property is immediate deterministic finality: once committed, a block cannot be reversed without an attacker controlling more than one-third of all BFT replicas. Finality latency L is bounded by the network round-trip time multiplied by a small number of phases, typically achieving $L < 1$ second in well-configured enterprise networks. The critical limitation is $C(n) = O(n^2)$: $for\ n = 100$, PBFT generates approximately 10,000 messages per consensus round, rendering the protocol impractical for large validator sets. Hyperledger Fabric accordingly targets validator sets of 3–15 nodes in typical deployments.

4.5.2 Tendermint and HotStuff

Tendermint marries PBFT-inspired safety guarantees with a stake-based Sybil resistance mechanism, enabling use in semi-public contexts where validators are known but not fixed by administrative fiat. Its two-round voting process — prevote and precommit — provides deterministic finality with a rotating proposer. The Cosmos ecosystem has demonstrated BFT-class finality with validator sets of up to 150 nodes at throughput figures of 1,000–10,000 TPS, while maintaining $C(n) = O(n^2)$ communication complexity. HotStuff represents the current state of the art in BFT protocol efficiency. Where PBFT requires $O(n^2)$ messages per round, HotStuff's use of threshold signatures reduces this to $O(n)$ — a dramatic improvement enabling BFT-class consensus for significantly larger validator sets. The proposer aggregates votes using a threshold signature scheme, broadcasting a single quorum certificate rather than relaying individual votes. This enables throughput T > 10,000 TPS with finality latency $L < 1$ second. Aptos and Sui blockchains, both descendants of Meta's Diem project, report sustained throughput exceeding 160,000 TPS under controlled benchmark conditions. For a validator set of n = 1,000, HotStuff generates approximately 1,000 messages per round, where PBFT would generate 1,000,000 — a three-order-of-magnitude reduction, making large-scale BFT deployment genuinely feasible.

4.5.3 Fault Tolerance Thresholds Across BFT Variants

Across all BFT protocol variants discussed in this section, the fundamental fault tolerance bound $f < n/3$ — derived from the Byzantine Generals Problem in Section 4.1.1 — governs safety. The differences among PBFT, Tendermint, and HotStuff lie not in their fault tolerance threshold but in their communication complexity, validator set scalability, and the mechanisms by which they achieve the $2f + 1$ quorum required for safe commitment. Hyperledger Fabric configurations typically set f = 1 with n = 4; Tendermint deployments often set f ≤ 50 with n = 150; and HotStuff-based systems have demonstrated practical operation with f ≤ 333 and n = 1,000. The selection of appropriate n and f for a given deployment is as much a governance decision as an engineering one.

4.6. EMERGING AND HYBRID CONSENSUS PARADIGMS

4.6.1 Proof of Authority and Proof of Elapsed Time

Proof of Authority replaces computational or economic Sybil resistance with identity-based accountability. A pre-approved set of named, identity-verified validators produce blocks; their public identities serve as proof of legitimacy, and the threat of reputational damage, legal liability, and exclusion provides the incentive for honest behaviour. PoA achieves T of 1,000–5,000 TPS with deterministic finality L < 1 second, and is particularly well suited to private enterprise consortia and regulated industries: supply chain management networks, healthcare data exchanges, and financial market infrastructure. The fundamental trade-off is analytically clear: PoA achieves high T, low L, and strong accountability by abandoning the permissionless and pseudonymity that many regard as essential to blockchain technology. Proof of Elapsed Time, developed by Intel for Hyperledger Sawtooth, leverages trusted hardware to implement a fair-lottery block production mechanism at minimal energy cost. Each node requests a randomly generated wait time from a Trusted Execution Environment (Intel SGX), with the node receiving the shortest wait time earning the right to propose the next block. PoET achieves

approximately 1,000 TPS with low energy consumption. Its significant structural limitation is dependency on Intel SGX: documented side-channel attacks, including Spectre and Meltdown variants targeting SGX enclaves, represent a supply-chain and hardware dependency that substitutes a distributed software trust model for a centralised hardware trust model.

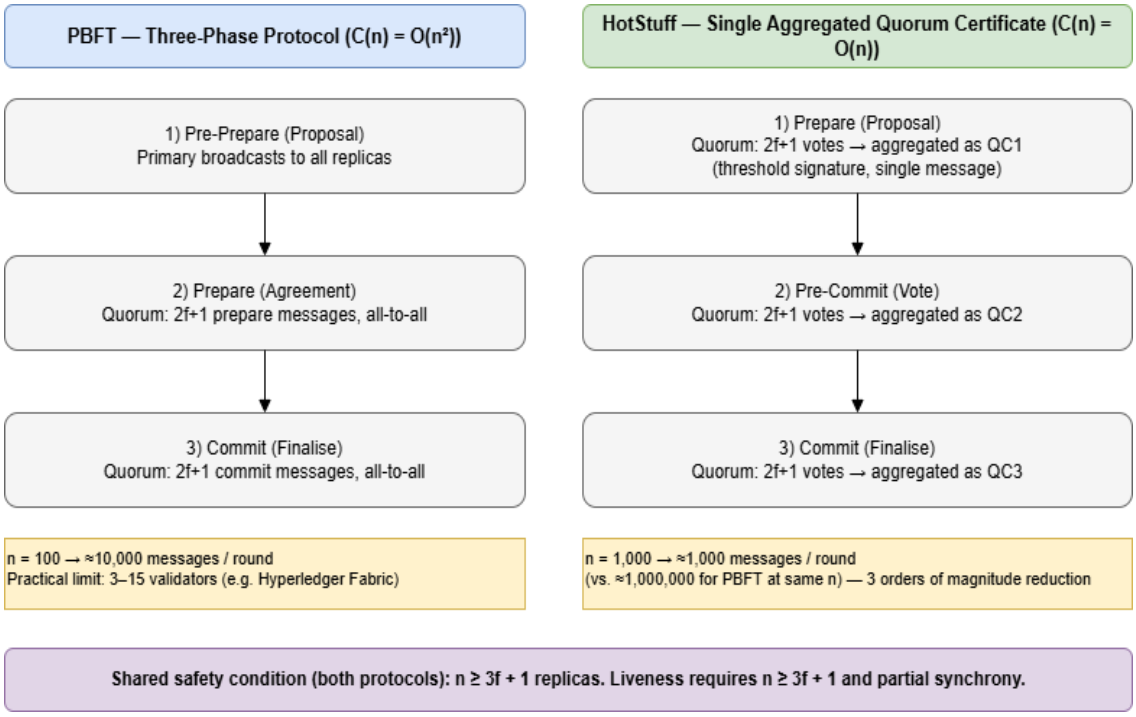


Fig-5 PBFT vs HotStuff

4.6.2 Proof of Reputation and Proof of Contribution

Proof of Reputation (PoR) extends PoA by introducing a dynamic, algorithmic measure of validator trustworthiness rather than static identity verification. In PoR implementations such as the GoChain network, each validator is assigned a reputation score computed from a weighted combination of historical transaction validation accuracy, network uptime, and staking commitment; validator selection probability is then proportional to reputation score rather than stake alone. This addresses a recognised weakness of both PoA (binary inclusion/exclusion) and PoS (stake concentration rewarding wealthy validators): reputation serves as a continuously updated signal of demonstrated reliability. Proof of Contribution or Proof of Importance, implemented in the NEM/Symbol ecosystem, extends this principle by weighting validator selection by a composite importance score incorporating transaction volume originated, network connectivity, and vesting time — effectively rewarding active participation in the network's economic activity rather than passive token hoarding. The fault tolerance properties of both mechanisms are analogous to classical PoS, with the additional complexity that the reputation and importance metrics themselves become attack surfaces: adversaries who can manipulate scoring inputs gain disproportionate consensus influence.

4.6.3. Proof of History and Tower BFT: The Solana Architecture

Solana's consensus architecture combines Proof of History (PoH) and Tower BFT. Proof-of-history (PoH) is not a consensus protocol, rather it is a cryptographic clock whose working is based on verifiable delay function. It creates an unforgeable timestamp sequence through

hashing of the output to create a continuous chain of inputs Each of the timestamps that get added into the PoH chain provides a cryptographic proof of the order in which each of those events occurred. Thus, validators are able to reach an agreement on the ordering of events without having to exchange their timestamps.Through utilizing the PoH clock as a shared and trusted time source, communication overhead is significantly reduced in coming to agreement about the ordering of blocks, enabling throughputs above 65,000 TPS under benchmark conditions and 2,000 to 5,000 TPS as of 2025 under real-world workloads.The operational experience of Solana presents an instructive case study: Seven instances of downtime between September 2021 and February 2023, including 17 hours of downtime in September 2021 and 7 hours of downtime in June 2022, arising from failures in validator coordination at very high transaction volumes. Such examples demonstrate the difference between benchmarking performance and reliability.

4.6.4 DAG - Based Approaches and Layer 2 Scaling

Directed Acyclic Graph-based consensus architectures represent the most radical structural departure from the linear chain model. In a DAG-based system, transactions are organised as individual nodes in a directed acyclic graph, and each new transaction is required to reference and validate a certain number of previous transactions. IOTA's Tangle requires each issuing device to validate two previous transactions, distributing validation work across all participants. Hedera Hashgraph combines a gossip-about-gossip communication protocol with virtual voting to achieve asynchronous BFT consensus with formally proved safety under the standard $n \geq 3f + 1$ bound and claimed T > 10,000 TPS. However, one must remember that security proofs of DAG-based blockchain systems are considerably less developed than those of chain-based blockchain systems, and many throughput claims were not independently verified under adversarial conditions.Layer 2 designs embody the solution that many leading blockchain platforms have chosen in order to reach the trilemma limit. Instead of seeking to maximise decentralization, security and scalability simultaneously on the protocol layer – an impossible task, as proven in Section 4.1 – Layer 2 designs delegate the responsibility of executing transactions to a second layer while relying on the underlying Layer 1 consensus algorithm for security Optimistic rollups (Arbitrum, Optimism) bundle the transactions off-period for fraud proofs, resulting in T of 2,000 – 4,000 TPS that are 10 – 100× cheaper than Layer 1. Zero-Knowledge.chain and upload compressed data and a purported state root to Layer 1 with a seven-day

Layer2 Architectures rollups (zkSync Era, Polygon zkEVM) include a cryptographic validity proof along with each state transition bundle, allowing immediate validation without any challenge period and resulting in T of 2,000 – 20,000 TPS. State channels (Lightning Network, Ethereum payment channels) allow parties to conduct any number of transactions off-chain, uploading only the initial and final states to Layer 1 – resulting in an effectively unlimited T between the parties involved at near-zero latency but the requirement of remaining online for all participants.

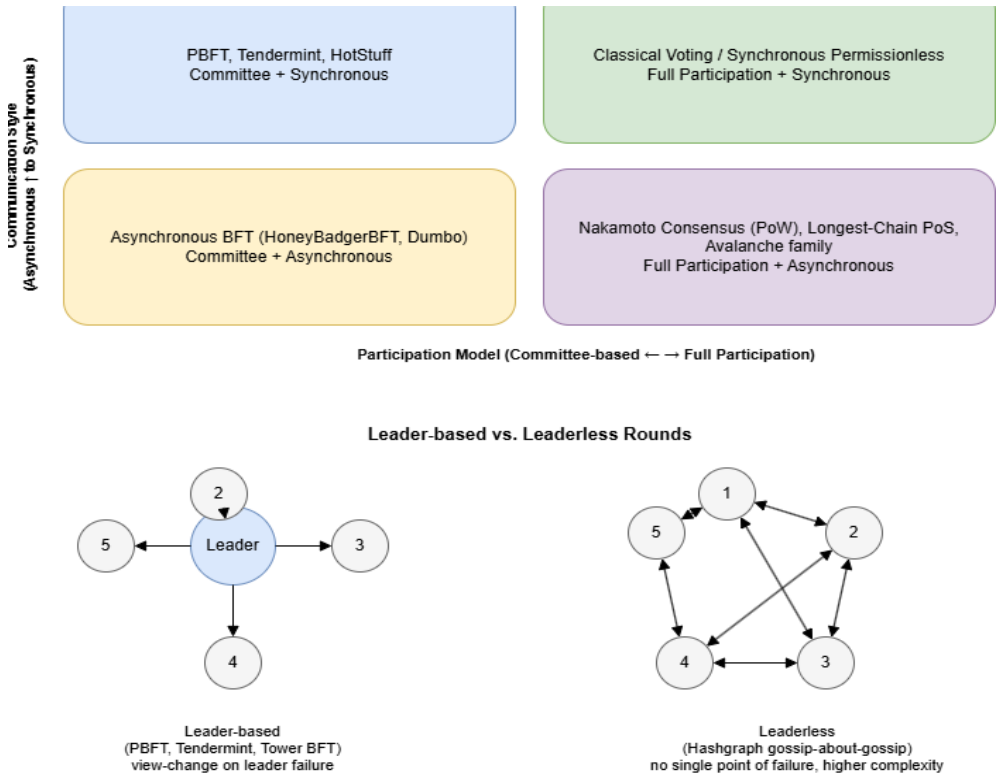


Figure-6. Layer2 Architectures

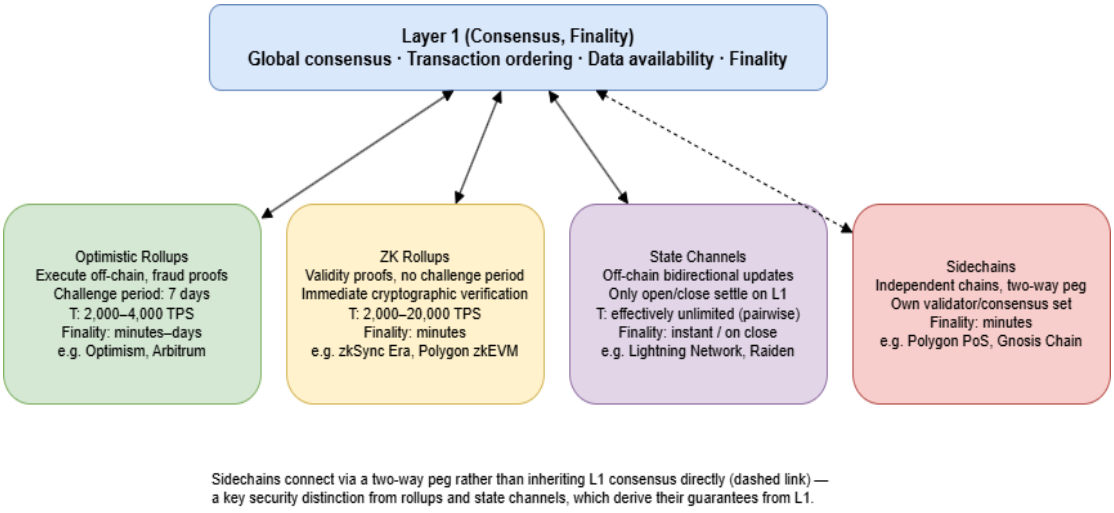


Figure-7 Trilemma Comparison Chart

4.7 COMPARATIVE ANALYSIS OF CONSENSUS MECHANISMS

4.7.1 Performance, Throughput, and Finality

Performance and finality are analytically inseparable. The distinction between probabilistic and deterministic finality is critical for application design. In PoW systems, confidence in a transaction's permanence grows incrementally with each additional block: six Bitcoin confirmations (approximately 60 minutes) provide a probability of reversal below 10^{-10} assuming an honest majority. For PBFT-class protocols, L is bounded by network round-trip

time multiplied by three protocol phases, typically L < 1 second in enterprise LAN environments. The Ronin bridge hack of March 2022 — a $625 million theft — exploited the period between probabilistic and deterministic finality in a cross-chain bridge design, illustrating concretely why the choice of finality model is a security-critical engineering decision rather than an abstract preference. The practical significance of finality latency extends beyond raw performance metrics to shape the economics of blockchain-integrated financial systems. Settlement risk — the probability that a transaction considered confirmed is subsequently reversed — carries a real cost in financial markets, where participants must post collateral or margin to cover the exposure window between probabilistic confirmation and true irreversibility. For PoW chains, this settlement risk window is non-trivial: Bitcoin's six-confirmation heuristic represents an industry convention rather than a formal guarantee, and the precise probability of reversal at any confirmation depth is a function of the total hashing power distribution that changes continuously. ZK rollups with cryptographic validity proofs represent the strongest achievable settlement guarantee at scale: the validity proof is

4.7.2 Security Guarantees and Attack Resistance

Security is a multidimensional property whose components must be individually evaluated. For PoW, the principal attack vectors are 51% attacks, selfish mining (profitable at 33% hash share under favourable propagation conditions), and eclipse attacks at the network layer. For PoS, the principal vectors are long-range attacks, validator cartelisation, and Nothing-at-Stake exploitation. For BFT protocols, the principal vectors are coalition attacks by correlated validators and denial-of-service against the proposer in leader-based designs. The fundamental observation that applies across all mechanism families: the security of a consensus mechanism cannot be evaluated in isolation from the incentive alignment of its participants. A protocol formally secure under its stated assumptions may nonetheless fail in practice if the economic environment creates systematic incentives for deviation — as the Terra/Luna collapse and EOS cartelisation dynamics demonstrate.

4.7.3. Design Patterns Across Consensus Mechanisms

A systematic review reveals four recurring architectural patterns. The first is the committee versus full-participation division: BFT protocols restrict validation to a small committee of authenticated participants, achieving low L and deterministic finality at the cost of limiting n and decentralisation, while PoW and classical PoS permit full participation, sacrificing deterministic finality for open membership. DPoS, PoR, and PoI represent hybrid positions concentrating block production in a small elected committee while permitting broad economic participation.

The second pattern is leader-based versus leaderless rounds: PBFT, Tendermint, and Tower FT designate a rotating proposer per round — a single point of failure whose replacement requires a view-change protocol that is the most complex and attack-prone component of much designs. Leaderless designs such as Hashgraph eliminate this vulnerability at the cost of increased protocol complexity and less mature security proofs. The third pattern is synchronous versus asynchronous communication assumptions: classical BFT protocols assume synchrony or partial synchrony, enabling strong finality but failing under real-world network partition; asynchronous BFT designs provide safety under pure asynchrony at the cost of liveness,

constrained by the FLP impossibility theorem. The fourth pattern is gadget-based versus monolithic protocols: Ethereum's Casper FFG is a finalisation gadget sitting atop LMD-GHOST, offering modularity at the cost of complexity; monolithic designs such as HotStuff integrate the entire consensus protocol in a single coherent mechanism,offering tighter security proofs and simpler failure mode reasoning.

4.8. ALGORITHMIC TRUST: A CRITICAL CONCEPTUAL FRAMEWORK

4.8.1 Trust as a Computational Construct

Classical social-scientific understanding treats trust as a subjective cognitive state — a disposition toward vulnerability grounded in expectations about another party's behaviour. The central idea behind blockchain technology is that computational systems governed by rules could create the functional equivalent of such a disposition in the absence of all those social, personal, and institutional means through which trust has been created in the past. This is trust without trustees, and this needs to be thoroughly analyzed. There are three criteria for a consensus algorithm to be capable of generating trust. However, it turns out that such safety is somewhat contingent, because the security of the digital signatures that underlie Bitcoin, Ethereum and almost all other blockchains is based on the difficulty of the discrete logarithm problem, which can be solved in polynomial time by a quantum computer equipped with Shor's algorithm. The first post-quantum cryptography standards were finalized by NIST in 2024, namely CRYSTALS-Kyber as a key encapsulation mechanism and CRYSTALS-Dilithium as a digital signature scheme. Migration towards post-quantum signatures for blockchain consensus layer protocols is still ongoing: there has not been a successful migration to the post-quantum protocol in any major public blockchain, and the window where a blockchain can be breached through quantum computers before it migrates is a very catastrophic gap in the cryptographic foundations of algorithmic trust.

Table 1 — Comparative Analysis of Consensus Mechanisms

Mechanism	TPS (approx.)	Energy Profile	Finality Type	BFT Threshold	Decentralisation	L2 Ready
Proof of Work (PoW)	~7 (Bitcoin)	Very High	Probabilistic	51% hash majority	Very High	Ltd.
Proof of Stake (PoS)	~15–45 (Ethereum L1, base layer)[1]	Very Low	Hybrid (Casper FFG)	≥⅓ stake	High	Yes
Delegated PoS (DPoS)	>3,000 (EOS)	Very Low	Near-instant	21-node set	Low (oligarchic)	Yes
Leased PoS (LPoS)	~1,000 (Waves)	Very Low	Probabilistic	≥⅓ stake	Medium	Prtl.

PBFT / Hyperledger	~1,000–10,000	Low	Deterministic	n ≥ 3f + 1	Low (permissioned)	N/A
Tendermint / HotStuff	~10,000+	Low	Deterministic	n ≥ 3f + 1	Medium	Yes
DAG (Hashgraph/IOTA)	~10,000+ (claimed)	Very Low	Async BFT	⅓ Byzantine	Variable	Prtl.
Proof of Authority	~1,000–5,000	Minimal	Deterministic	>50% validators	Very Low	Yes
PoH / Tower BFT (Solana)	~65,000 (benchmark)[2]	Low–Med.	Optimistic + BFT	Supermajority	Medium	Emrg.

Second, the economic incentive structures of the protocol must remain aligned with honest behaviour across the full economic lifecycle of the network. The 2016 DAO hack — in which $60 million of Ether was drained through a reentrancy vulnerability — illustrates the gap between consensus-layer security and application-layer vulnerability: the consensus mechanism performed exactly as specified, but the code implementing the DAO contract failed to instantiate the properties claimed for it. The subsequent Ethereum hard fork, in which the community voted to reverse the hack by altering the canonical chain history, illustrates that when stakes are sufficiently high, the human governance layer overrides the algorithmic consensus layer — a reminder that algorithmic trust is situated within, and ultimately subordinate to, social and political trust. Third, the code implementing the protocol must actually instantiate the properties claimed for it in its specification — a condition that formal verification methods attempt to provide but have not yet been applied at scale to production consensus implementations.

4.8.2 Governance, Fork Mechanics, and the Political Economy of Consensus

Governance is not peripheral to consensus; it is the meta-consensus layer determining whether the algorithmic consensus layer continues to function as intended. On-chain governance — exemplified by Tezos's formal amendment process and Polkadot's OpenGov system — encodes protocol change procedures directly into smart contracts, enabling validator sets to propose, vote on, and ratify protocol upgrades without hard forks. This provides a transparent, auditable change mechanism but concentrates governance power in proportion to stake, introducing the same cartelisation risk documented for DPoS. Off-chain governance — the model used by Bitcoin and Ethereum — relies on informal social coordination and is more resistant to stake-concentration capture but less legible and subject to opaque influence by well-resourced stakeholders. Fork coordination mechanics illustrate the governance challenge concretely. The Bitcoin/Bitcoin Cash split of August 2017 and the Ethereum/Ethereum Classic split of 2016 both arose from irreconcilable disagreements that consensus-layer mechanisms did not resolve — they enforced, by enabling each faction to operate a separate canonical chain. Enterprise blockchain deployments must therefore embed validator set governance in legally enforceable consortium agreements rather than relying on algorithmic mechanisms alone.

4.8.3. Implications for Secure Intelligent Ecosystems

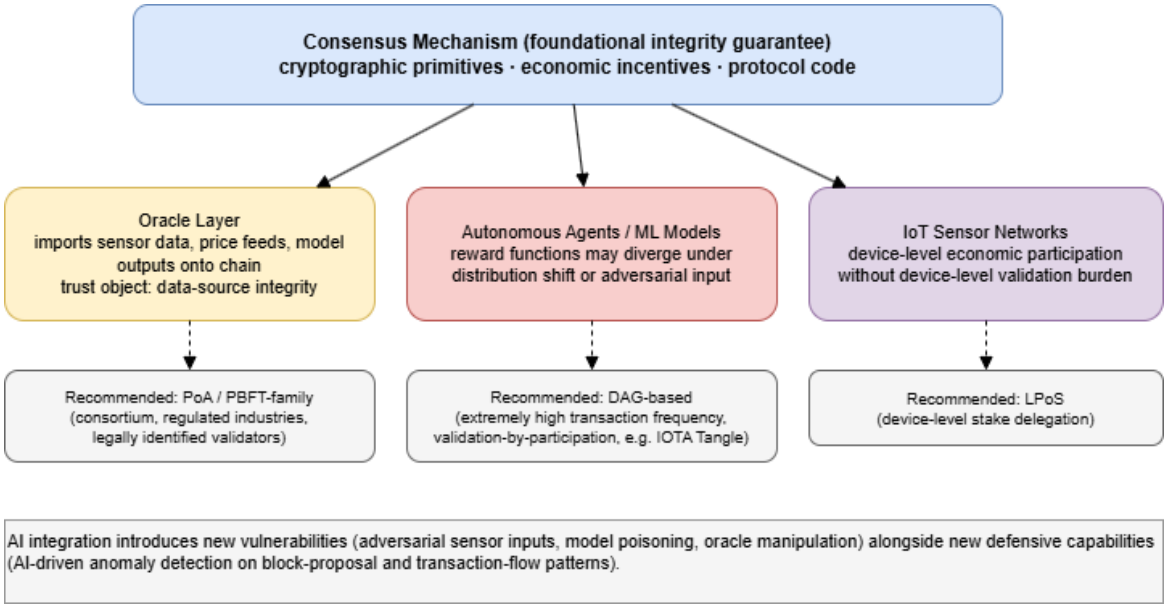


Figure-8. Trust Layering IoT/AI

The deployment of blockchain consensus mechanisms in intelligent ecosystems — environments characterised by deep integration of IoT sensor networks, machine learning models, autonomous agents, and data-sharing platforms — amplifies both the potential benefits and risks of algorithmic trust. A practical deployment framework requires matching protocol architecture to deployment context. The algorithmic trust framework developed in this chapter identifies three distinct trust objects to which trust is relocated from identified human actors when a blockchain consensus mechanism is deployed: the cryptographic primitives, the economic incentive structures, and the code implementing the protocol. In intelligent ecosystem deployments, each of these objects of trust gets more and more complex. The trust in the cryptographically-secure layer must include the trust in the interaction between the blockchain and its external environment, which are the oracle mechanisms bringing the sensor data, price feed and model results into the blockchain. The trust in the intentions of the agents must recognize the possibility of inconsistency between those intentions and the safety assumptions of the blockchain, due to distribution shift and adversarial inputs.Additional layers of trust require additional levels of security architecture, where the consensus protocol acts as the base-layer integrity service for other layers. In the case of resource-constrained IoT deployments, the optimal solution would be the LPoS, as it allows incentivizing but not validating the devices.DAG-based architecture is well-suited for frequent transaction IoT deployment scenarios, like IOTA Tangle. For consortium deployments in regulated industries, PoA and PBFT-family protocols provide the appropriate properties: deterministic finality L < 1 second, legally identified validators, and governance structures mappable to existing regulatory frameworks. The integration of AI creates both new vulnerabilities — adversarial inputs to sensor data, model poisoning attacks, manipulation of oracle systems — and new defensive capabilities, as AI-driven anomaly detection can identify unusual patterns in block proposal behaviour or transaction flow that might indicate an ongoing attack.4.8 Practical Deployment Considerations Network latency is the primary determinant of achievable finality latency L in practice, and its effects are frequently underestimated in benchmark conditions. BFT protocols

achieving L < 1 second in LAN environments may achieve L of 3–10 seconds in geographically distributed deployments, as multi-round message exchanges compound network propagation delays. Optimistic rollup settlement is dependent on Ethereum Layer 1 block time, multiplied by the required number of confirmations for considering the state root as a trustworthy one – 12–51 minutes in case of high-stakes settlements. System architects should be aware of production-level latency budgeting rather than benchmarking when determining whether the protocol fulfills application SLA needs.

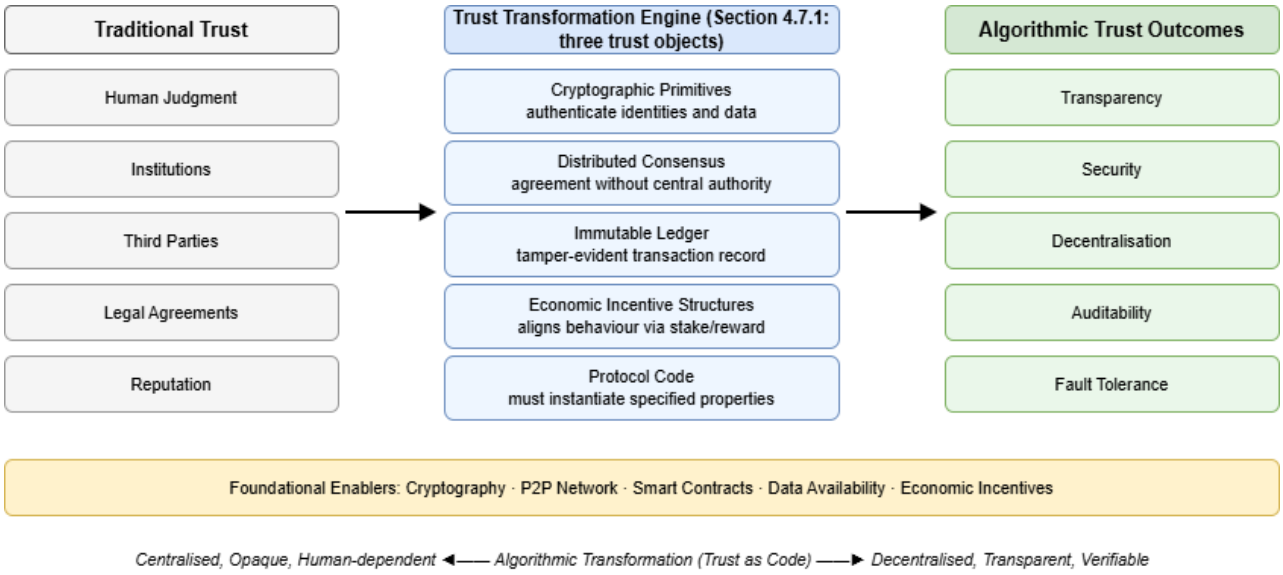


Figure-9. Trust Transformation

Figure 9, Economics of trust transformation validators: Hardware costs, staking demands, and Maximal Extractable Value (MEV) significantly impact the security and decentralization of the consensus protocol. Hardware requirements for Ethereum validators are a modern server processor, 16-32GB RAM, more than 2TB high-speed SSD and broadband Internet connection; the staking requirement of 32 ETH (approximately $80,000-$120,000 according to 2024/25 prices) makes a wealth barrier for individuals who want to become validators. It leads to the fact that the majority of stakers are concentrated in staking pools (Lido, Rocket Pool, Coinbase), which account for most of the staked ETH, while PoS aimed to mitigate the situation but was unable to resolve it. MEV arises from the possibility of the validator to choose and arrange the transactions' inclusion order in the block, which makes additional value through front running, sandwiching liquidity provision, and arbitrage of the public mempool. It is estimated that total MEV captured from the Ethereum network from 2020 to 2024 surpasses several billions of dollars. The separation of the function of building blocks from the function of proposing blocks, known as Proposer/Builder Separation (PBS), is currently the established engineering solution, somewhat reducing MEV centralization by the introduction of competition in block building. But PBS itself leads to its own centralization forces, with the emergence of specialized block builders that capture most of the MEV revenue and thus form an economic oligopoly in the block-building space despite a decentralized validator set. Client diversity is an often-overlooked factor in consensus system robustness. In case the majority of validators uses the same client implementation, a bug in the client can lead to simultaneous failures in the whole set of validators, even if the consensus algorithm itself is perfect. The diversification of clients used for the Ethereum 2.0 is a direct reaction to this problem, with the goal of not having a single

client taking more than 33% of the validator set. In enterprise blockchain systems, one faces a similar problem: Hyperledger Fabric systems using only a single ordering service implementation suffer from organizational single point of failure.

4.9 CONCLUSION

This chapter has pursued a single central argument: consensus mechanisms are not technical details but sites where fundamental decisions about trust, power, and accountability are encoded into computational form. trust landscape theory. Comparative analysis shows that blockchain trilemma is an inherent characteristic of the design space and is compliant with the constraints imposed by Byzantine Generals Problem and CAP theorem. It is not possible to design a mechanism which would incorporate the best aspects of decentralization, security, and throughput simultaneously. The trade-offs identified in the mechanisms comply with the constraints: Proof-of-Work offers maximal decentralization and security but does not provide T; BFT mechanisms offer maximum T and deterministically final transactions but do not offer decentralization; Proof-of-Stake and its variations are somewhere in the middle where extremes are defined by economic incentive compatibility and governance. The algorithmic trust model from Section 4.7 is a robust tool of analysis of any consensus mechanism. The process of turning human trust into computation, cryptography, and economics, each having their own way of failing. The forks of history, the vulnerability of cryptographic protocols in the presence of quantum computing, and the oracle problem for AI systems are but a few instances where the process of turning trust is failing and invisible to the eyes of the original dreamer of the trust transfer process. Some of the most intriguing solutions to the problems raised by the trilemma frontiers – layer 2 scaling techniques, Byzantine Fault Tolerance Protocol design inspired by Hotstuff, and post-quantum cryptography changes – are so advanced that they matter, but certainly not solved yet.

These patterns of design including committee vs. full participation, leader vs. leaderless turns, synchronous vs. asynchronous premises, gadgets vs. monolithic architecture - allow to define the language for describing future innovations and understanding their trade-offs before practical implementations. Chapter 5 continues from here, discussing cryptographic primitives, such as hash functions, digital signatures, and zero knowledge proof systems, that are used as the machinery of algorithms behind consensus protocols.

CHAPTER 5

ARTIFICIAL INTELLIGENCE FOUNDATIONS: MODELS AND LEARNING PARADIGMS

Contributed by: Subhadip Kowar, Shreya Basu, Souvik Biswas, Mousumi Das

5.1 INTRODUCTION AND THE TRAJECTORY OF COMPUTATIONAL AUTONOMY

Over the years, the understanding of computer science has changed significantly. In the past, programmers thought it was possible to resolve complicated problems by means of very complicated algorithms that were similar to procedure instructions and thus could be performed by the computer. A lot of work was done on coding all the rules of the specific system, taking into account all kinds of possible cases. However, the technique was successful if the systems that were considered were simple and comprehensible. These days distributed systems comprise a very complex network of edge nodes creating enormous volumes of heterogeneous datasets that simply cannot be dealt with using standard software engineering approaches of manual coding.

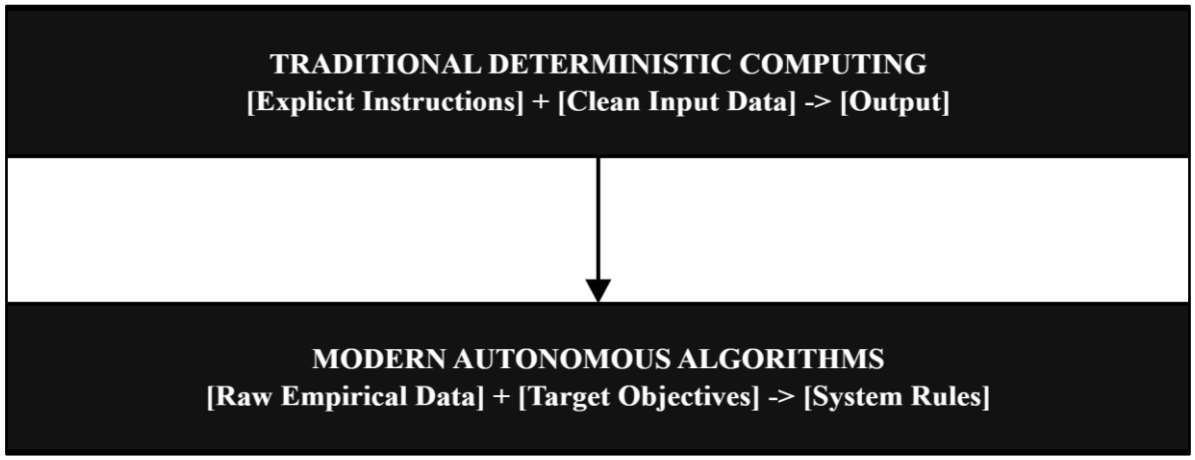


Figure 1. Traditional Computing vs. Modern Autonomous Algorithms

The advent of artificial intelligence is the beginning of a new paradigm change in the core design principles in software engineering. Instead of using human specification of domain logic through deterministic conditionals, self-running algorithms today turn the computing paradigm upside down by using optimization techniques. Software design is able to create the needed set of operational rules based on the empirical distribution of the underlying data. It is crucial in order to solve the problem of dynamic distributed systems. Nowadays, deterministic

software designs face a risk of underperformance under stress situations. Thousands of edge devices creating the non-stationary, high-throughput data make it impossible to manually craft rules. Manual human oversight is unable to keep up in such a data environment because of high volumes and velocity of the data. In cases when the network infrastructure faces a DDoS attack or other anomalies, it is impossible to maintain the latency needed for human intervention or the patch installation. It needs intelligent processing that will be able to detect deviation, analyze the risks and apply the threat mitigation measures within the range of microsecond time frames. For instance, after receiving a certain signal from the computer, the system should process the signal to check its safety and in case if the signal is unsafe, then the system should perform the threat-mitigation process in real time. To create such systems, the knowledge an understanding of computational cognition and learning is necessary. This issue is covered in the current chapter.

5.2 FOUNDATIONS OF KNOWLEDGE REPRESENTATION:

However, prior to processing information about the environment, discovering any implicit structure and making a decision in an intelligent system, it is needed to convert information from its physical representation to a properly formed computational model. The process of representing real-world information in a computational form is considered as the main challenge for the domain. This topic has been under intensive study in the context of computer science for many years and there have been two major theoretical paradigms developed to solve this problem: symbolic model and connectionism. The symbolic theory of cognition, which has been used extensively in the beginning of artificial intelligence studies, assumes that cognitive functions are based on manipulations of symbols. In this model, the knowledge is represented with declarative networks, semantics and production rules. As an example, the automatic controller of the network can use logic rules while analyzing state vectors: if the node has an abnormal flag at an open port and the traffic from this node exceeds some threshold, then isolation is required. Despite the fact that such an approach provides auditability and clarity of the model, it has a number of restrictions in the real world.

5.2.1 The Symbolic Paradigm and Rule Based Logic

Symbolic artificial intelligence refers to the application of computations that involve symbolic manipulation of the objects, ideas, and activities that occur in the world around us. The symbol can refer to either the actual object or idea. Rules, propositions, and computations are used by computers to make decisions. The information contained in the computer can be easily interpreted because the information contained in the computer is represented in a manner that humans can understand. For example, a computer program that keeps a network safe may use a rule that says what to do if something happens. Symbolic artificial intelligence uses these declarative rules to make automated operational decisions. This deterministic deduction process is formally expressed as a first-order predicate logic rule in (5.1):

$$\forall x(Node(x) \wedge Compromised(x) \rightarrow Isolate(x)) \qquad (1)$$

Let x represent any processing entity in the architecture. This setup makes it clear that if the entity is recognized as a node in the network and is at the same time determined to be

compromised, an isolation process needs to begin. This arrangement presents numerous advantages for engineering since the logic involved is completely understandable. With these good things systems that only use symbols have some big problems when we use them in the real world. They have three limitations:

- **The Monotonicity Constraint:** Old style logic systems have trouble changing their conclusions when they get incomplete or conflicting information.
- **The Combinatorial Explosion:** When we add variables and rules to a system it gets really slow because it has to look through so many possibilities. This uses up all the computer's memory and processing power.
- **The Semantic Gap:** We cannot easily convert data from sensors like network traffic or sound waves into simple symbols without losing important details. The software setup has trouble understanding the context of the systems. The symbolic systems are not good at handling this kind of data.

5.2.2 The Connectionist Paradigm and Distributed Representations

Connectionist Paradigm is considered an AI paradigm that imitates the features of the human brain. Unlike conventional methods in which AI problems are solved using explicit programs or discrete symbols that store the data in this model, knowledge is built up in the form of a big network of small computational units called artificial neurons that work in conjunction through interconnections with some weight values similar to biological neuronal nets. The fundamental concept of this model is distributed representation. According to connectionism, none of the neurons holds all the information within the information unit. The many-to-one ratio between the neurons and information makes the network flexible and the system becomes more robust since the knowledge is not located at one place but spread in the network. In case any of the neurons fails or works improperly, the network still works well since all the missing parts do not have all the information together.

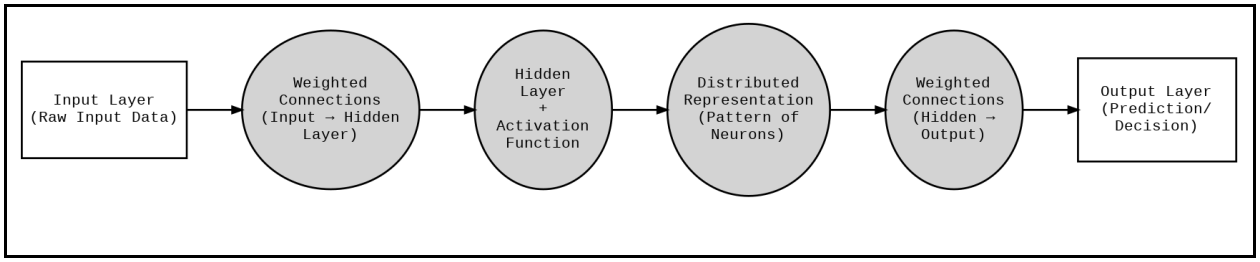


Figure 2. Distributed Data Representation in Connectionist Networks

One of the biggest advantages of distributed representations is that the network learns its own features, rather than having people spell them out. In traditional artificial intelligence programs, the task of choosing the important feature is done by a human expert, which is difficult and restrictive. Not a problem with connectionist systems. The neural net, which has been developed based on large training sets, is capable of recognizing connections and useful information even if it is hard for humans to recognize them. It helps considerably improve the efficiency of the connectionism method in areas such as image recognition, speech recognition, and language processing since it can recognize important characteristics by itself. Another

difference is the fault tolerance feature which suggests that these systems are quite robust. They do not have centralized data and it means that the destruction of some elements does not disrupt the whole working process of the network. Although destruction of neurons and synapses causes its inefficiency, it usually does not disrupt its functioning and leads to accurate results. This quality is especially useful in practice because data are often distorted and the hardware is imperfect. Connectionist networks have another ability to improve their performance, when they change the value of the weights during the training. This is done using gradient descent backpropagation etc. This is done as new data is fed into the system for training. Each of the mechanisms contributes towards decreasing errors step-by-step towards the targeted output during successive cycles of training. Another benefit that comes from generalization, the ability to use acquired knowledge in new and unseen situations – is seen too. Connectionist systems learn the inherent structure and relations within training samples, but not just remember them. So, for example, when a system is trained to distinguish between handwritten digits, it actually learns the structure of digits, but not their samples. Therefore, it is able to recognize other handwriting samples which have never been seen before. This ability forms the basis for applying connectionist models to real-life tasks.

5.3 TAXONOMIES OF COMPUTATIONAL ADAPTATION AND LEARNING PARADIGMS

The process of adaptation of internal parameter settings within an intelligent system highly depends on the type of learning algorithm upon which the system is based. They can be classified depending on the structure of the data processing pipeline, mathematical properties of the feedback loops, and the way the information about the environment is fed to the optimization module. In order to build a solid foundation for higher educational curricula, they are grouped into three main families: supervised learning, unsupervised learning, and reinforcement learning. Each approach deals with different computational tasks, defined by their own set of compromises between sample complexity, computational complexity, and other limitations.

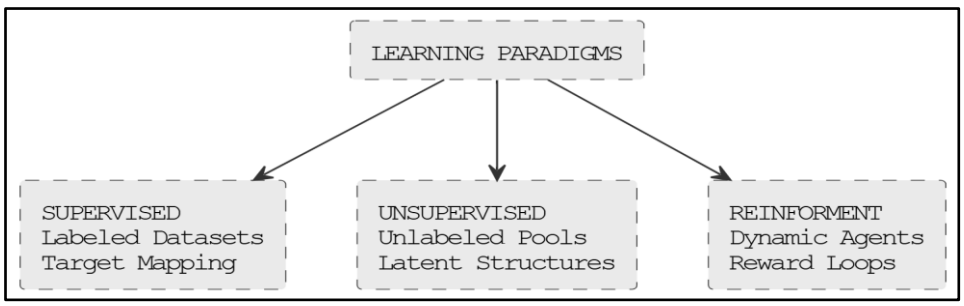


Figure 3. Taxonomy of Core Machine Learning Paradigms

5.3.1 Supervised Learning Frameworks

In order for the supervised learning process to be effective, there must be access to a structured training set, which includes the input vectors, as well as the correct labels for those inputs.

Mathematically, the aim of this process is to model the underlying function f, which maps complex inputs to target outputs. This fundamental target relationship is formulated as:

$$y = f(x) + \epsilon \quad (2)$$

In this case, x is the high-dimensional feature vector while y is the ground truth target label and ε stands for the error term associated with the environment in which data is being collected. In any training session, the prediction machine produces an estimate of the output represented by $\hat{y}$. The explicit loss function $L(y, \hat{y})$, measures the structure error continuously as the difference between the estimated value and the actual target label. The optimization process involves computing the derivative of the loss function with respect to the network parameter at all network topologies in order to reduce the error. The optimization process involves calculating the partial derivatives of the loss function across all network topologies to minimize overall error. The mathematical rule representing the continuous parameter update via the gradient descent algorithm is given in (5.3):

$$w \leftarrow w - \eta \nabla L(w) \quad (3)$$

In this case, the variable w stands for the weight matrix of the model, η is the learning rate parameter responsible for the step size of the optimization process, and ∇L(w) represents the gradient of the loss function concerning the weight matrix. Supervised learning optimization problems fall into two primary categories depending on the type of mathematical properties of the variable that is being optimized.

- **Classification:** The model outputs a class boundary for the input variable, which could be something like detecting malicious packets in networks or making classifications in the field of medicine.
- **Regression:** The model outputs numeric values for the input variable, like predicting latency time or traffic pattern trends in a network infrastructure.

5.3.2 Unsupervised Learning and Latent Structure Extraction

Unsupervised learning algorithms work on datasets that lack predefined labels or any external means of error correction. In the machine learning algorithm, there is no guiding signal from the outside to adjust the parameters. Instead, the algorithm needs to look at the data matrix and discover patterns, clusters, or distributions within itself. Such techniques are quite significant for data exploration and data preprocessing in a secure environment. This happens through two mathematical approaches, namely:

- **Clustering Based on Coordinates of Matrices:** k-means clustering algorithm, for instance, partitions data sets to form distinct clusters based on the proximity in the multidimensional feature space. Points that have a small distance from each other are placed in the same clusters, thus forming relationships without human intervention.
- **Dimensionality Reduction through Manifolds:** Technological systems often yield high dimensional matrices filled with a lot of statistical noise and redundancy. Mathematically, this can be illustrated using techniques such as PCA, which transform the high dimensional feature space into lower dimensional coordinate spaces. As this transformation occurs, the

noise and statistically correlated variables are stripped away to preserve the maximum variance.

5.3.2.1 The Data Preprocessing Pipeline

In academia, the datasets used in the process of algorithmic analysis possess a high level of structured data. In contrast, the data gathered from automatic systems that work in actual environments is known for its noisiness. To use unsupervised learning algorithms to discover any hidden structure in data or form clusters, one needs to go through a thorough data preprocessing stage on the input data. Using the data acquired directly from sensors as an input to the machine learning algorithms will yield unsatisfactory results since it is known in the field of computer science that bad input equals bad output. One of the major problems in data preprocessing is the existence of statistical noise. There might be failures of sensors themselves; sensors located in smart buildings and biosensors on people could move randomly without having anything to do with the phenomena being measured. In distributed edge networks, problems with data come from the removal of noise. There is poor connectivity and occasional power supply interruption in remote nodes. Disconnection of a node leads to a cessation of data transmission and creates a gap in the continuous sequence of matrices. It is important to solve this issue because some algorithms rely on having complete data. Yet, deletion of the whole row means the loss of valuable contemporaneous data coming from active sensors. And the method of imputation can help with this problem. It involves estimation of missing data according to the historical patterns of the particular sensor. Problems with standardization of data are also found in the modern self-managed system. Data from several sources is processed at the same time. For example, the Internet of Things network has a measurement of CPU temperature in degrees centigrade, network latency in milliseconds, and bandwidth in megabytes. These features have various scales of measurements. And if input this data in a coordinate-based clustering algorithm without standardizing it, then the result would be biased toward larger numbers. Feature scaling is used in order to reduce the bias due to different magnitudes of features. Different features are converted into one numerical scale through mathematics processes, usually through normalization in [0, 1] range. Hence, the contribution of features is equal in unsupervised algorithms, meaning that small changes in temperature will weigh as much as large changes in latency. Also, one needs to consider the issue of categorical variables. Neural networks work with numeric inputs, and, in practice, there are lots of categorical variables in data. After detecting whether the distributed node is active, idle or compromised, it is necessary to transform this information into numeric form so that the neural network could use it. Development of such a preprocessor takes lots of time and efforts including constant data cleanup and conversion into the proper format in order to make the input understandable by the learning algorithm.

5.3.2.2 Deep-Dive Analysis of The Data Preprocessing Pipeline

Development of data pre-processing pipeline is among the core challenges for developing autonomous edge networks. While the analysis of the data is usually carried out using the assumption of having perfect input matrices, the reality is far from that. Real-life data is always noisy and nonlinear.

Statistical Denoising and Signal Isolation

IoT industrial and biomedical monitoring systems operate under considerable interference in their physical and electromagnetic environment. Industrial edge devices used in a factory are subjected to various influences like vibrations, changing temperatures, or fluctuating power that affects measurements. Interference takes the form of statistical noise that might make it hard to notice any trends in the data set. In cases when a standalone model performs analysis of data without pre-processing, optimization layers might interpret noise as valid signal which will cause overfitting. In order to prevent this situation, filtering is used in the data flow in the edge. The moving average filter smoothens sudden changes in the data set, while low-pass filter reduces the influence of electromagnetic noise in the form of high frequency waves. However, the use of filters should be done carefully, as choosing wrong filters might cause additional delays in processing the data which negatively affects the system. Milliseconds delay in processing high speed data flows, which appear in autonomous cars and drones, might prevent anomalies detection.

Imputation Models for Missing Data

Data inconsistency in edge networks might result from latency, packet losses, and power interruptions. Disconnection of the edge device from the network or entering of it in low-power mode causes inconsistency of the data matrix. This is critical problem in the cases when neural model relies on feature vector. Exclusion of the rows of the data matrix with missing value might involve omission of other useful information collected by the working sensors.



Figure 4. Sequential Stages of the Edge Data Preprocessing Pipeline

The data pipelines make use of imputation strategies that help in filling in missing values through the use of past experience along with inter-sensor relationships between neighboring sensors. Whereas linear interpolation is effective when there is steady state of the parameters involved, in case of complicated and time variant parameters, advanced prediction strategies should be used. This is where spatial temporal imputation strategy helps.

Multi-Scale Standardization and Feature Normalization

In modern autonomous systems, several variables are simultaneously captured from various sources. Think of an urban infrastructure network where a multi-agent system takes into consideration the CPU temperature measured in degrees centigrade, latency measured in milliseconds, and throughput measured in gigabytes per second. The variables have very different ranges. In cases where the information is used in distance-based clustering, such as k-means, the variables that have larger absolute values get more weight, such as the huge increment in the value of latency to 2,500 ms. This implies that any changes in the smaller ranges can be overlooked. Feature scaling is one way of dealing with such problems. It involves bringing all the input variables to a common range, usually between 0 and 1.

5.3.3 Reinforcement Learning and Sequential Decision Processes

Reinforcement learning shifts the focus of computation from passive data banks to dynamic environments that allow interaction. Reinforcement learning is a framework in which the agent

tries to optimize efficiency of functioning in an unknown environment by experimenting and receiving feedback in the process. Mathematical formulation of the connection between these two concepts relies on the Markov decision process. Interaction involves alternating states, actions, and scalar feedbacks. The autonomous agent evaluates the current state S of the environment, chooses an appropriate action a according to the agent's internal policy settings, and then changes to a new state s' of the environment. In the process, it receives a scalar value of reward R from the environment as a result of the transition. The main task of the agent is not to get the rewards right now but to change the policy so that the agent gets the most rewards overall when we look at a really long period of time. The primary objective of the reinforcement learning algorithm is the determination of the optimal policy of operation of converting the states of the environment into actions such that long-term accumulated rewards are maximized. The problem of optimization involves the challenge of finding the right balance between feedback and consequences. An autonomous system should often forego the immediate reward for the sake of avoiding states of greater risk and penalty in the future. From the mathematical point of view, the task lies in the consistent balance between exploring and exploiting the state-action space. The agent needs to explore new areas of state-action space for gaining rewards and exploit the existing high reward trajectory. The agent is trying to maximize the expected gain of the agent over a long period of time. This sequential optimization methodology forms the basis for algorithms behind industrial robotics, distributed scheduling, and autonomous traffic routing.

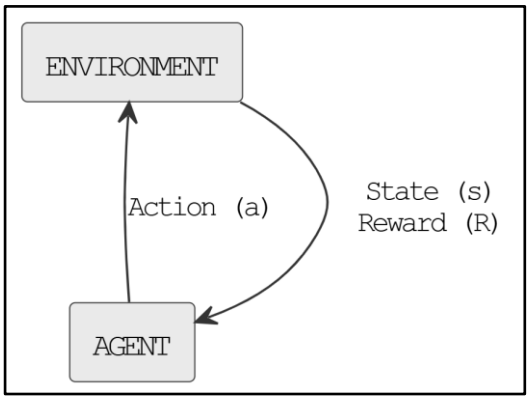


Figure 5. The Feedback Loop in Reinforcement Learning

5.3.4 Multi-Agent Systems and Urban Digital Twins

The limitations of reinforcement learning agents become visible when scaling from isolated machines to infrastructure-level. In order to understand how algorithms solve complicated tasks, it would be useful to introduce the notion of urban digital twin. An urban digital twin is a perfect virtual copy of a physical urban environment that is created using continuous streams of data produced by sensors scattered throughout the streets, the power grid, and the transportation network. Unlike a mere map, such digital cities act as dynamic simulators and can help study how various parameters affect the urban system. However, a city has an incredible number of components and it is simply unrealistic to manage all the city processes through a single reinforcement learning agent. The size of the state and action spaces would be so great that an optimal policy cannot be computed even by the best computer on Earth. This problem is solved using transition from single-agent reinforcement learning to multi-agent

systems. Hundreds or thousands of autonomous agents operate in this kind of system inside the digital twin. Each of those agents manages some portion of the infrastructure – one of them regulates a traffic light on the intersection, while another one controls energy supply for a certain block of a hospital. All those agents operate in one and the same environment and that causes a new difficulty because the action of one agent affects the environment for others. The case where the traffic controller agent tries to optimize green light times to reduce local congestion leads to higher traffic flow downstream to the junction, which requires the next set of agents to adjust to the sudden change in their environment. The coordination between the agents will have to happen without the help of any central governing authority. This is highly possible depending upon how the rewards are designed. The rewards which encourage maintaining traffic flow at one particular intersection can lead the agent to reroute the traffic to different roads and reduce the efficiency of the whole city. Thus, the reward structure will need to be such that it can encourage the cooperation of the agents. Agents can be trained in such a way that they not only optimize themselves but also the entire system. For example, an agent can receive a small punishment for causing heavy traffic delays elsewhere in the city. In the virtual setting, agents learn through trial and error, and thus find solutions to urban traffic issues using a multi-agent system. Tasks such as rerouting traffic after being disturbed or load balancing in the traffic network can be assigned to agents.

5.4 ADVANCED ARCHITECTURES AND DECENTRALIZED LEARNING

With the increasing complexity of computations, there will be many limitations that centralized methods of training would face. The intelligent models that exist in today's era need to have high capacity, low latency, and proper data management. To achieve these criteria, it becomes very important to move forward from basic learning algorithms and use advanced neural networks.

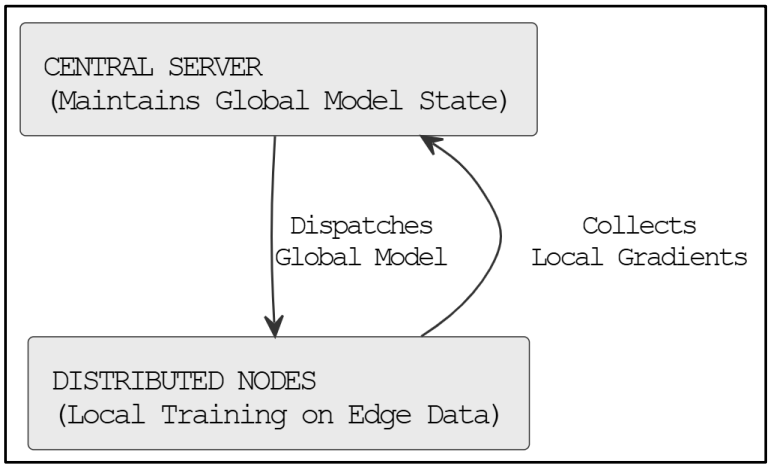


Figure 6. Decentralized Federated Learning Architecture

5.4.1 Deep Neural Networks and Transformers

The idea of deep learning can be described as the evolution of connectionist models. It involves the existence of many hidden layers which serve to process information in the stages between the input and output layers. As a result, the model is capable of analyzing information using the

technique of feature extraction, where the first layers detect such basic features as edges, frequencies, or gradients, and then the next layers use the information to build more abstract representations. With the invention of Transformers came an important revolution in working with sequential data. In traditional models, which worked in a sequential way, there was a significant computational problem which made training very difficult and hard to parallelize. The mechanism of self-attention used in transformers solves this problem by allowing calculating mathematical relations of all tokens in the sequence simultaneously, instead of sequentially. It makes it possible to establish relations at long distances regardless of the distance quickly. Thus, a parallel training algorithm becomes possible.

5.4.2 Federated Learning Paradigms

The concept of centralized machine learning is fundamentally wrong since it assumes that edge devices require transmitting their data from the cloud for achieving the optimal model. This is not practical in current distributed systems because of issues related to insufficient bandwidth, high latency, and data privacy breach. The concept of federated learning provides a way out through the use of the right infrastructure. This approach brings about a paradigm shift in the traditional training of models because it shifts the learning to the point where data is located locally. The optimization works like this:

- The coordinating server distributes a copy of the current global model structure among the active edge devices.
- Each device runs local iterations based only on its own privately held data.
- The local devices do not exchange any data after the local iterations; rather, they transfer just their unique weight or gradient updates to the central server.
- The coordinating server combines all of these local updates into one global model template.

The coordinating server combines these decentralized updates into a single global model template. This parameter aggregation is mathematically performed by the Federated Averaging (FedAvg) algorithm outlined in (5.4):

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \qquad (4)$$

In the above-mentioned functional equation, K stands for the total number of active edge nodes involved in the training block.n_k is the number of data samples present on the node k, whereas n is the sum of all the data samples present in the whole distributed network. w_{t+1}^k is the set of distinct weights calculated at node k in its training phase. This way of doing things helps to reduce the cost of communication and makes the process safe, from a point of failure. The methodology also follows the laws related to data privacy very closely. The data privacy laws are something that this methodology takes into account.

5.4.3 Application Paradigm: Wearable Bio-Sensing and Digital Phenotyping

In order to illustrate how much decentralized learning architectures can be helpful, some applications addressing particular problems connected with modern healthcare technology will be provided. For example, there exist biosensors that constantly record various physiological and behavioral information. It is related to the concept known as digital

phenotyping, which is associated with the analysis of human-machine interaction and the use of the wearable sensor information for the creation of the whole picture of people's psychological and physical health status. There are machine learning algorithms, which can recognize certain behaviors using such metrics as physical activity, sleep, heart-rate variability, and data about human-machine interaction. In this way, it can be possible to diagnose mental illnesses earlier. For example, the absence of physical activity along with the lack of regular sleep schedule means that a person is depressed. From the perspective of centralized learning paradigms, it is problematic as the information necessary for tracing the psychological condition of a person is highly personal and confidential; therefore, it poses certain threats regarding cyber-attacks. The theory of federated learning appears to be appropriate in resolving this problem. Contrary to techniques involving uploading the health information of users to a central model, the model gets uploaded to the user's device. The bio-sensing unit acts like a standalone edge node in the whole system. It makes use of the generic model for identifying behavioral patterns and training them using the data of the particular user. After training the generic model locally through the user's device, the wearable device determines the modified parameters of the model and sends them to the coordination server mathematically. The coordination server combines the structural changes of different wearables and modifies the generic model accordingly. As a result, the system gradually learns how to identify behavioral patterns of people with different characteristics. Besides, the sensor data will stay confidential and available only on the user's device.

5.4.4 Hardware Limitations and IOT Edge Computing

With the transition of AI to edge computing and integration into the physical world, an extremely realistic test follows. Algorithms assume plenty of computational power; however, the use of federated learning and multi-agent systems involves running computations on edge devices. Edge computing means running algorithms on the Internet of Things (IoT) devices like smart cameras, environmental sensors in industry, and wearables. On one hand, on-device computation solves the problem of bandwidth and privacy issues, on the other hand, it gives rise to a completely new problem associated with hardware limitations. It is not possible to run a model trained on servers on a tiny microchip. The key issue here is that there is no sufficient memory. Modern deep learning models, especially with many layers of neurons, need plenty of RAM (in gigabytes) to keep parameters of an algorithm in memory. In turn, an average edge node in IoT devices can provide just a few megabytes of memory. Trying to compute neural network on these small microcontrollers is doomed to fail because of serious memory limitations which make it impossible to hold matrices and perform necessary computations. Alongside memory issues, energy restrictions must be met in the design as well. The centralized data center constantly consumes energy from the power supply network but at the same time uses an extremely efficient liquid cooling system that cools down the heat produced by its processors. However, edge computing devices work differently. Most IoT sensors use wireless connection and are battery-operated. The amount of energy needed for constant operation connected to matrix multiplications is beyond the limits of these batteries. For example, a self-contained monitoring point that runs out of battery charge after just three days because of the huge neural network model will stop operating the entire system. Besides, these compact chips do not have any fans for cooling and, thus, intense calculations may result in overheating and

deterioration of chip performance and even its physical damage. Another major limitation refers to the processing speed. Cloud servers use dedicated processing units that are designed for parallel computations, whereas common edge devices use microcontrollers with low clock rates. Machine learning on an edge point is associated with sequential processing of data or even its limited parallelism that leads to high latencies. For example, an autonomous drone that should detect the object instantly cannot afford ten seconds of the calculation period. To overcome the difference between the considerable software needs and the restricted hardware resources, it is necessary to change the models physically. The task is done through the process of model quantization, where the precision of the network weights is reduced. To be more precise, 32-bit floats are converted into 8-bit integers. In doing so, the memory used by the model is significantly reduced, as well as its processing speed, although at the expense of lowering prediction accuracy a bit. Another technique utilized for reducing the computational load is network pruning. In the case of the fully trained deep neural networks, many connections between neurons have low weights and do not affect the result much. By pruning these connections, one can create a sparse model that takes up less space and uses less energy. By means of such manipulations, artificial intelligence can be incorporated into very limited devices called edges.

5.4.5 Security Vulnerabilities and Threat Mitigation

There have been quite some changes witnessed in the security landscape due to the shift from centralized cloud architectures to decentralized edge computing. The federated learning architecture ensures privacy protection by keeping the data in edge devices but faces numerous inherent architectural challenges. It does not make sense any more to assume that all the nodes along with the coordinating server are completely trustworthy. The primary vulnerability of the federated learning architecture is visible when there is a case of gradient inversion attack. In conventional federated learning architectures, training occurs in the edge devices where the gradients are shared to the central server. There was an assumption that the gradients shared would give out nothing about the data used. But malicious nodes or even the coordinating server can use the gradients to generate fake data. In order to protect the privacy from any such invasion, techniques like differential privacy have been used where gradients are added with random noise before transmitting to the central server. Gradient compression has also been used as one such technique that accomplishes the above stated task. Not only is it important to protect the system from any such privacy invasions but at the same time, it becomes crucial to adopt some security measures to defend against any sabotage attempts through the process of data poisoning. In data poisoning, an attacker tries to manipulate the local training data so that the global model suffers because of it. One common way to accomplish such data poisoning is via label flipping, in which labels are changed for some training examples with an intent to confuse the decision boundary. There are many ways by which the model can be attacked via backdoors and exhibit some abnormal behavior under certain circumstances. In case the adversary manages to gain access to an edge node, there is no need to poison the input anymore as it would directly attack the model.The prominent example of the model poisoning attack is the replacement of the global model by simply scaling the malicious update by such a big factor so that one can completely replace the global model in a single round. The straightforward averaging cannot be used in order to protect the architecture from malicious updates. Even if

there is one corrupted gradient, simple averaging will result in corruption of the whole system. That is why it is important to use robust Byzantine aggregation mechanisms such as trimmed mean or geometric median. These methods allow analyzing statistical distribution of the incoming updates and rejecting those which look like an outlier based on this analysis.

5.4.6 Expanded Security Analysis and Defences

As autonomous frameworks move from centralized cloud architectures to decentralized edge environments, the attack surface changes dramatically. Security models can no longer assume that edge nodes or the central coordination server are completely secure.

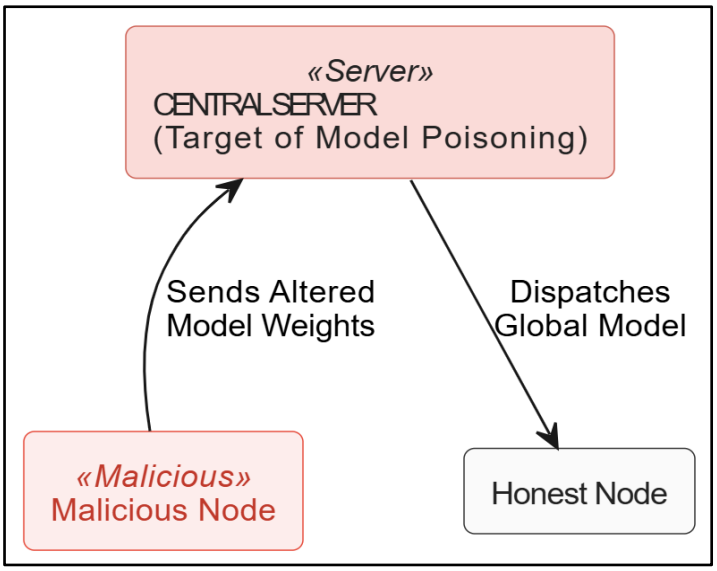


Figure 7. Architectural Overview of a Model Poisoning Attack in Federated Learning

Gradient Inversion Attack and Privacy Protection

While federated learning is privacy-preserving because the data is stored locally and only the gradient vectors are transferred, the mechanism still is vulnerable to gradient inversion attacks. A gradient inversion attack is aimed at capturing the gradient vectors sent from the edge device. Using an optimization technique that employs artificial input vectors, an attacker manipulates the input in a way that the generated gradients match the captured vectors. This vulnerability can be overcome using differential privacy methods.

Data And Model Poisoning and Byzantine Averaging Techniques

Data and model poisoning remain potential threats in decentralized systems. The data poisoning occurs when malicious training data is added to the local dataset due to label flipping. Model poisoning means modifying the local weights before their transmission with the purpose of poisoning the global model. To protect against these kinds of threats, Byzantine fault-tolerance techniques like trimmed mean and geometric median filtering are used at the server side.

5.5 STRUCTURAL SYNTHESIS AND COMPARATIVE ANALYSIS

Computational complexity is among a number of factors that must be taken into account when designing modern day autonomous agents because of the constraints present within the hardware and architecture of computing machines. There is no learning paradigm or framework that solves all possible problems for all possible computing situations through a single framework. In order to assist in choosing frameworks for intelligent distributed environments, the following table summarizes the analysis of the various learning paradigms discussed within this chapter.

Table 5.1: Comparative Analysis of AI Learning Paradigms

Type of the Framework	Primary Optimization Objective	Computational Bottlenecks	Core Structural Vulnerabilities
Symbolic AI	Manipulation of explicit logical tokens and formal rules	Exponential search space expansion	Complete system failure when encountering unmodeled states
Supervised Learning	Minimization of loss function $L(y, \hat{y})$ via gradient descent	High cost of matrix multiplication during backpropagation	High dependence on massive labeled datasets and susceptibility to distribution shifts
Unsupervised Learning	Discovery of latent geometric structures and data distributions	High dimensional coordinate calculations	Sensitivity to initialization parameters and convergence to sub optimal local minima
Reinforcement Learning	Maximization of cumulative long-term reward expected return	Exploding state action spaces in unconstrained environments	Reward function hacking and high sample inefficiency during early training exploration
Federated Learning	Decentralized weight optimization via localized updates	Network latency during global parameter aggregation cycles	Vulnerability to gradient inversion attacks and malicious local data poisoning

The future development of autonomous infrastructures depends a lot on the architectural hybridization of the technologies. Whereas deep layered connectionist networks are still prevailing in feature extraction procedures, it is their black box nature that represents considerable threats in secure environments. The integration of the verifying capabilities of symbolic rules-based methods and the continuous parameters of the connectionist networks suggests an essential shift in paradigms in computer science. Using them in connection with federated optimization algorithms, it is possible to create intelligent nodes capable of

implementing complicated decision-making procedures without having any data stored in the cloud.

CHAPTER 6

CONCEPTUAL FRAMEWORKS FOR CONVERGENCE OF BLOCKCHAIN AND AI

Contributed by: Dr. Abhijit Paul, Dipanjan Das, Manoj Kumar Rabidas

6.1. INTRODUCTION

The internet has changed a lot over the past fifteen years. Two big things have made a difference. The first one is Blockchain. It started in 2008. People needed a way to send money to each other without a man. Now, Blockchain is used for things. It helps make sure that things are true. It keeps records in a way that lots of people can see. It is really useful. The other big thing is Artificial Intelligence. It has been around since the 1950s. At first, it just followed rules. Now it can do a lot more. It can look at lots of data. Find patterns. It can even make guesses. Create new things. These new things are as good as what people can do. Earlier, different groups of people worked on Blockchain and Artificial Intelligence. They did not actually talk to each other. Companies paid them to solve problems. Blockchain and Artificial Intelligence were like two worlds. The gap between these technologies is getting smaller fast. Experts believe that the global Blockchain market will increase from around $564 million in 2024 to more than $2.4 billion by 2030. This means the Blockchain market will grow by more than 27 percent every year. The McKinsey Global Institute says, artificial intelligence applications could add up to $13 trillion to the economy by 2030. As Blockchain and artificial intelligence get better and more people start using them, people are wondering what will happen when the secure system of Blockchain is combined with intelligence's ability to learn and make decisions. The early results are looking good. Blockchain can give intelligence a record of where the training data comes from. This is important, for Blockchain and artificial intelligence because it helps them work together better. Blockchain and artificial intelligence are both important technologies. When Blockchain and artificial intelligence work together they can do some cool things. It can also help manage model decisions in a way and reward people for honest participation, in learning. AI can make Blockchain networks better by making them faster and more secure. It can help spot transactions and enable smart contracts to make decisions on their own. Blockchain and AI can work together to make each other stronger. The combination of Blockchain and AI can build a cool thing.

Blockchains are good for security, and AI is good at learning. People are really looking into Blockchain and artificial intelligence. This is a deal because Blockchain and artificial intelligence can be combined to make a tool that helps us figure things out and learn new things. People still do not really understand how the Blockchain and artificial intelligence work together. Most of

the things that have been written about the Blockchain and artificial intelligence are about areas like healthcare, supply chains, the Internet of Things and finance. People will learn about the Blockchain and artificial intelligence. How can they work together? The rest of this chapter is divided into four parts. The first part is Section 3 it is about Blockchain. This section of the chapter looks at Blockchain. How Blockchain was made. The people who made Blockchain wanted to achieve things, so this section goes back to that. It also looks at the areas where Blockchain is used today. Then it talks about the problems that Blockchain has. These problems with Blockchain mean that Blockchain cannot work on its own as a system Blockchain needs something to work properly. Section 4 is similar to Section 3. It is about artificial intelligence. This section looks at how artificial intelligence has changed over time, from using symbols to using lots of data to learn. It also talks about the uses of intelligence and the problems that artificial intelligence has. The problems with intelligence include things like intelligence being hard to understand and people being able to mess with the data that artificial intelligence uses. Artificial intelligence also has problems with security and privacy issues. Section 5 is where Blockchain and artificial intelligence come together. This is a part. It uses a framework called AdaptChain to demonstrate how Blockchain and artificial intelligence can work together as one system. Blockchain has some things about it and these things can help fix the problems that artificial intelligence has. At the time, artificial intelligence has some good things about it and these things can help fix the problems that Blockchain has. So, Blockchain and artificial intelligence can help each other out. This chapter has explained the joint role of Blockchain and AI, which are both efficient on their own. Blockchain helps with the security of data and documents, and makes secure transactions. AI helps with predictions and recommendations. When Blockchain and AI are combined, the best of both technologies is achieved. Though there are a lot of articles written about it, the main focus is to show the effectiveness of using AdaptChain model. AdaptChain is the model which brings together two things: Reinforcement Learning, and Blockchain technology. This proposed model helps AI and Blockchain technology work together in a way that's fair and good for everyone. AI systems have some problems, like users don’t know how they make decisions and sometimes their decisions may not be fair. On the other hand, Blockchain networks have problems too like being too rigid using much energy. The AdaptChain model solves all these defined problems. The model helps people trust each other, makes sure that everyone is responsible, which leads to trust, transparency and fairness.

6.2. RELATED WORK

The Blockchain and AI are being looked at together often now. We want to see how the Blockchain and artificial intelligence work when they are used together not on their own. There are some studies about the Blockchain and artificial intelligence that we should consider. These studies look at the Blockchain and artificial intelligence in various ways. One study looks at the field of the Blockchain and artificial intelligence and how it has changed over the years. Another study looks at the ways the Blockchain and artificial intelligence can be used together. A third study looks at how to make the Blockchain and artificial intelligence work together, in a system. Bhumichai *et al.* did a study of sixteen important articles about the Blockchain and artificial intelligence. They found fourteen things that keep coming up like keeping data safe keeping data private and making decisions automatically. The people who did this study also looked at how the field of Blockchain and artificial intelligence has changed over time. They said there were three times in the history of Blockchain and artificial intelligence. The first time was when

Blockchain and artificial intelligence were developing separately. The second time was when people started combining Blockchain and artificial intelligence. The third time was when people started using the technology of Blockchain and artificial intelligence in real life. This study is helpful because it shows that Blockchain and artificial intelligence are good at things. It also shows that there are still some problems with Blockchain and artificial intelligence that need to be solved, like how to make sure the information is true, how to make the system bigger, and how to keep data private. These problems with Blockchain and artificial intelligence will come up again when we talk about AdaptChain, and Blockchain and artificial intelligence. The Blockchain and artificial intelligence are important to this discussion about Blockchain and artificial intelligence. The Blockchain and artificial intelligence have some problems that need to be solved. We will talk about these problems, with Blockchain and artificial intelligence.

Choi and Kim take a look at how AI and Blockchain work together. They suggest a way to organize research on this topic based on how they interact. When AI is used to improve Blockchain they group the studies into three areas: The protocol part, where AI helps make consensus and mining better like using learning strategies to mine effectively. The data part, where AI finds patterns and detects cheating in transactions and ledger data such as finding Ponzi schemes on Ethereum. The application part, where AI makes contracts and decentralized apps better. When Blockchain is used to improve AI they find four types of studies: Building learning platforms on Blockchain. Keeping records of model training. Updates on Blockchain. Sharing data safely to develop models. Using Blockchain to govern AI making sure it is fair and can be checked. Their study supports an idea that this chapter also explores: Blockchain gives AI the ability to be checked and trusted without an authority and to show if data has been changed, which AI does not have on its own. In return, AI gives Blockchain the smartness it lacks to make decisions on its own. Choi and Kim are open about the limits of studies, saying most just use Blockchain to store data, not to work together with AI. They try to go beyond this limit with their way of organizing and a model that we will talk about later. The two surveys mentioned earlier show how wide the field is. Mandinyenya and Malele give an example of what a combined system can look like in real life, echoing other converged frameworks that pair reinforcement learning with Blockchain consensus and coordination. The AdaptChain framework was created using a research method. It has three parts. The Artificial Intelligence part of AdaptChain uses a reinforcement-learning agent. The Blockchain part of AdaptChain helps with the agreement. Keeping track of things. The ethics part of AdaptChain is controlled by an organization. The reinforcement-learning agent in AdaptChain looks at things like how transactions take how long it takes for blocks to spread, how much AdaptChain can do and how much energy AdaptChain uses. It changes some settings, like block size and group members to make sure everything in AdaptChain is safe. This way, the AdaptChain network can adjust in time. The ethics part of AdaptChain uses some techniques to keep contributions to AdaptChain. A decentralized organization makes decisions about fairness in AdaptChain. This organization makes sure everything in AdaptChain is fair. They tested AdaptChain against three systems: BlockAI, AIChain, ChainML. AdaptChain did better in areas like adaptability, compatibility, speed and energy use, which is in line with other efforts to optimise Blockchain scalability.

AdaptChain worked well. It is a good thing for AdaptChain. The AdaptChain framework is very useful for things. The people who made AdaptChain tested it, and they are happy with AdaptChain. AdaptChain is a system, and it can be used for many things. However, the authors note that these results are from simulations. The prototype still needs to add some techniques to prevent bias. This is a limitation that the whole field has. People usually want to move rather than take the time to see if things actually work in real life. When you read these three parts together, you get a picture of a field that's very advanced from a technical standpoint but still

figuring some things out. The surveys in show that a lot of things are coming together, and this is happening in almost every major industry, and AI's own demonstrated capacity to solve previously intractable scientific problems shows why this convergence carries real stakes. AdaptChain shows that we can make this coming together work in a real system. What is still not well developed and what this chapter is trying to do, is explain why this coming together works at a basic level. We want to understand the design philosophy behind each technology, including AI's well-documented struggles with explainability and data integrity, and see what gaps each one leaves open. Then we want to show how these gaps fit together. The next two sections will do that, starting with Blockchain.

6.3. TRADITIONAL USE OF BLOCKCHAIN TECHNOLOGY

6.3.1. Origins and Core Design Philosophy

The story of Blockchain begins in 2008 when an anonymous individual or group operating under the pseudonym Satoshi Nakamoto published a whitepaper titled "*Bitcoin: A Peer-to-Peer Electronic Cash System"*. The core problem Nakamoto was trying to solve was surprisingly straightforward — how do two people transfer value over the internet without trusting a bank or any central authority to validate the transaction? The answer was a distributed ledger maintained by thousands of computers simultaneously, where any attempt to alter a record would be immediately visible to the entire network. The foundation of Blockchain design is based on three things: immutability, decentralization and transparency. When we talk about Blockchain design, the immutability property means that once you put data on the Blockchain it is there forever and cannot be changed or removed. The decentralization property of Blockchain design means that no one person or group is in charge of the network. The transparency property of Blockchain design means that everyone who is part of it can check every transaction on their own. The Blockchain design and its three properties are Immutability, decentralization and transparency, solved a problem with digital money, which is that someone could spend the same money twice. The best part is that the Blockchain design did this without needing some central authority to be in charge.

6.3.2. Traditional Application Domains

Over the years, Blockchain has really. It is used for a lot more than just digital money now. Blockchain is used in different areas. Some of the areas where Blockchain is used are explained below.

Cryptocurrency and Financial Transactions: The widely known use of Blockchain is for digital money that people can send to each other directly. Bitcoin showed that people could send and receive money over the internet without needing a bank. This was a deal because it let strangers trust each other with money. Ethereum took this idea further by adding contracts. These contracts are like agreements that automatically do things when certain conditions are met. They helped create a new world of financial services that are run by logic, not banks. Now, things that used to take days to settle with banks can happen in minutes. The best part is that there's a record of all these transactions. This makes it all transparent and trustworthy. Blockchain and digital currencies, like Bitcoin and Ethereum are changing the way we think about money.

Supply Chain Management: Global supply chains have a lot of people involved. Like the people who build things, the people who move things around, the people who check things at the border and the stores that sell things. It is really hard to keep track of a product as it moves from one person to another. Then something called Blockchain came along. It is like a book that lots of people can write in and nobody can erase what has been written. So, when a product moves from the person who made the parts to the person who buys it everything can be written down in this book. Big companies like Walmart and Maersk tried using Blockchain. They found that it really helped stop people from cheating and selling fake things. It also helped solve arguments, about what was shipped.

Healthcare Data Management: Patient records in healthcare are all over the place. They are stored in places like hospitals, clinics and diagnostic labs. These places use computer systems that do not work together. Blockchain technology provides a way to create a permanent health record. This record can be accessed from different places. It also keeps a record of every time someone looks at it. Some projects, like MedRec have used the Ethereum system. They looked at how patients can own their medical history. Patients can share this history with doctors in a way. The goal is to help patients control their medical information. They should be able to share it with doctors and other healthcare workers when needed. This can help improve healthcare. Some of these projects have shown promise. They have looked at how Blockchain can be used to keep records safe and secure. This approach could help solve the problem of patient records. It could make it easier for doctors and patients to work together. Projects, like MedRec are important. They are exploring ways to use technology to improve healthcare.

Smart Contracts: Smart contracts are super important for Blockchain, more than just for money. They are like computer code that lives on the Blockchain. This code automatically makes sure everyone follows an agreement when certain things happen. You do not need a lawyer, a notary or a bank, for this to work. Smart contracts are really changing how we do things. They help make agreements safe and fair. Blockchain makes sure smart contracts work correctly. Smart contracts are a part of Blockchain technology. Insurance claims, property transfers, escrow arrangements, and supply chain payments have all been explored as use cases where smart contracts reduce cost and eliminate the need for trusted intermediaries.

Digital Identity and Voting: Traditional digital identity systems rely on centralised databases that are single points of failure and regular targets for data breaches. Blockchain-based identity systems offer an alternative where identity credentials are stored in a decentralised manner and controlled by the individual, not the service provider. Similarly, Blockchain has been explored as an infrastructure for electronic voting, where the immutability of the ledger prevents vote tampering after the fact.

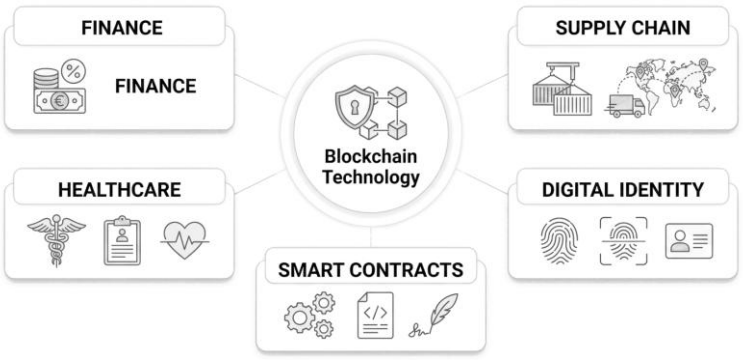


Figure 1: Traditional Blockchain Application Landscape

6.3.3. Inherent Limitations of Standalone Blockchain

It is easy to look at the list of applications above and assume Blockchain is a universal solution. It is not. When deployed in isolation, Blockchain has well-documented structural limitations that make it unsuitable for the kind of intelligent, dynamic systems that modern problems demand.

No Adaptive Intelligence: Blockchain executes fixed, pre-programmed rules — nothing more. A smart contract does exactly what its code instructs, in exactly the conditions its programmer anticipated. The real world, however, rarely cooperates with pre-written assumptions. Conditions evolve, edge cases emerge, and meaningful decisions often require contextual judgment that no static rule set can capture. Blockchain has no mechanism to learn from past transactions, detect shifting patterns, or update its own behaviour over time [4]. This is not a flaw that better engineering can fix — it is a fundamental characteristic of how the technology was designed.

Scalability and Computational Cost: Traditional Blockchain networks, particularly those using Proof of Work consensus, are computationally expensive and slow by design. Bitcoin processes approximately seven transactions per second; Visa, by comparison, handles several thousand. This performance gap makes Blockchain impractical for applications that require high-speed processing or real-time responsiveness — precisely the kind of performance that intelligent applications need.

***Inability to Process Unstructured Data*:** Blockchain handles structured transactional data well — numbers, addresses, hashes, timestamps. What it cannot do is interpret unstructured data: images, audio files, sensor streams, natural language text, or behavioural logs. It can store a cryptographic hash of an image as proof of its existence, but it has no understanding of what that image contains. As the world generates increasingly complex and unstructured data, this limitation becomes a significant ceiling.

Human Dependency in Decision-Making: Smart contracts must have every rule written in advance by human programmers. Situations that were not anticipated during design cannot be handled gracefully — the contract either fails, produces incorrect results, or halts entirely. This brittleness is a serious problem in dynamic, high-stakes environments where conditions change faster than code can be updated.

These limitations reveal that while Blockchain provides an unmatched foundation of trust, it lacks the cognitive layer needed for intelligent and adaptive decision-making — a gap that artificial intelligence is precisely designed to fill.

CAPABILITY	CORRESPONDING LIMITATION
Executes pre-programmed rules	Cannot adapt to conditions not anticipated in code
Immutable data storage	Errors or bugs are permanent and difficult to correct
Cryptographic security	Vulnerable to attacks if cryptographic keys are compromised
Distributed ledger	Requires high energy consumption for consensus (Proof of Work)
Transparent transaction history	Can reveal transaction details making full privacy difficult
Peer-to-peer network	May face network scalability and latency issues under high load

Figure 2: Limitations of Standalone Blockchain

6.4. TRADITIONAL USE OF AI

6.4.1. Origins and Basic Philosophy

Artificial intelligence, as a scientific field, can be dated back to the mid-twentieth century. Alan Turing's pioneering work in his 1950 article Computing Machinery and Intelligence asked the question, which defined further decades of research in this field: can machines think? The basic philosophy of first attempts at artificial intelligence relied heavily on symbolic reasoning; programmers manually created logical trees for solving problems and made rules for computer-based reasoning that worked in limited domains, but were too inflexible and rigid to handle real-life situations. What marked the transition to modern AI is a philosophical shift; instead of programming the rules, modern machine-learning systems derive rules and knowledge directly from the training data. Given a sufficient number of labelled examples, a neural network would learn how to recognize faces, how to translate languages, how to detect fraud, and how to make a prognosis of a disease without being instructed on how to do it. This learning process is powered by the combination of three factors: large amount of data, high-power hardware, and advanced algorithms. In practice, most of the AI currently in use belongs to the category of narrow AI, which refers to AI built to do a particular job really well, but which cannot extrapolate beyond it in any meaningful way. For example, an AI that can defeat world-class chess players is not necessarily able to read a chest X-ray.

6.4.2 Traditional Application Domains

Natural Language Processing: NLP includes the collection of technologies that enable machines to comprehend, process, and create human languages. The initial uses of NLP included search engines and spam filters. Recent advancements in NLP have made it incredibly powerful — language models can now summarize legal papers, translate between many languages instantly, analyze customer sentiments, and even create coherent content in just about any domain. Nearly all technological products have some form of NLP built into them.

Computer Vision: Computer vision is really good at helping machines understand what they see. This is one of the ways that artificial intelligence is being used. For example, computer vision can look at pictures and find cancer when it is just starting out. It is as good, at finding cancer as a doctor who specializes in reading these pictures. Computer vision is also used in cars that can drive themselves. These cars use computer vision to see the road and drive safely. Security systems use computer vision too to recognize people's faces and figure out who they are. Computer vision is used in different ways and it is very useful.

Recommendation Systems: Some of the uses of Artificial Intelligence so far are recommendation engines. Companies like Netflix, Amazon and YouTube use machine learning algorithms. These algorithms learn from users' actions. What they watch, click, search or buy. They make predictions about what a user will watch or buy next. This kind of personalization through machine learning is really hard to do. The recommendation engines help these companies give users what they want. Machine learning makes it possible to offer personalization. It is used by big companies to keep users engaged. Netflix, Amazon and YouTube are some examples of this. They all use machine learning in their recommendation engines.

Fraud detection and Cyber Security: Banks and financial institutions were among the first to use Artificial Intelligence (AI). They were also very eager to adopt this technology. Machine Learning (ML) models that use transaction data can help find patterns that might mean fraud is happening right now. These patterns are often too numerous and too complicated for a system that relies on a set of rules to deal with.

Healthcare and Drug Development: The use of Artificial Intelligence in healthcare has been a change. Googles DeepMind technology, which is called AlphaFold was able to figure out the shape of proteins something that biologists had been trying to do for a long time, for more than fifty years. This showed that Artificial Intelligence can solve tough science problems. Now doctors get help from computers when they need to diagnose patients and predict if their health will get worse. Artificial Intelligence is used in these computer systems to help doctors make decisions about patient care. Artificial Intelligence is really making a difference in healthcare.

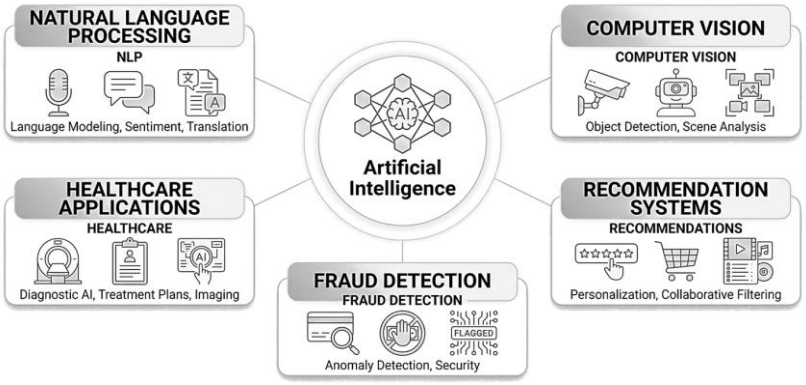


Figure 3: Traditional AI Application Landscape

6.4.3 Inherent Limitations of Standalone AI

The strengths of Artificial Intelligence are quite remarkable. However, there are some problems with the numbers, which become serious issues when we actually use AI in real life. These problems are not mistakes in the way AI is designed that can be fixed by updating it; instead, they are fundamental weaknesses of how AI is designed and are surprisingly similar, to the strengths of Blockchain technology. The AI strengths are impressive. Its weaknesses are concerning.

The Black Box Problem: Neural networks just cannot tell us why they make decisions. If a neural network decides to approve a loan or say a transaction is fake, or even suggest a treatment it cannot tell us what made it decide that way. The connection between what we put in and what comes out is buried in a number of numbers that no person can understand. This is an issue because neural networks cannot make things clear or explain themselves. This is a problem when we want to use this technology in fields where we need to know exactly how decisions are made. According to Omohundro intelligence systems that work towards certain goals without being clear about how they make decisions are not safe at all. We cannot check these decisions because there is no record of how they were made. Neural networks make it hard for us to follow what they do.

Data Integrity and Poisoning Attacks: The accuracy of an Artificial Intelligence model depends on the accuracy of the information it is trained with. If someone with malicious intentions can mess with this information before or while the model is being trained, which is called data poisoning, the model will make mistakes. These mistakes might not be easy to notice at first. There is no way in the normal Artificial Intelligence process to make sure the information used to train the model was not tampered with before it was used. The Artificial Intelligence model will only be as good as the information it is trained with so if the information is bad, the Artificial Intelligence model will be bad too.

Centralisation and Monopolisation of Control: The old way of making and using Artificial Intelligence is to have companies, government offices or research places do it. These places have the money and people needed to make Artificial Intelligence models work. The problem with this way is that it can be hard to say who is responsible when things go wrong. It is also hard to make sure the models are fair. This is because we do not always know how the models were made or who picked the information used to train them. Artificial Intelligence models can make mistakes. It is not always clear who is to blame. Zyskind et al. considered this issue and suggested Blockchain technology as the infrastructure for decentralised ownership of data in which people retain control over their data and don't give up their data to the centralised AI system for receiving a service.

Data Collection & Privacy: AI systems perform optimally when there is an abundance of data available to them. However, such an abundance of data, especially of personal and behavioral nature, can pose severe threats to privacy of individuals. While the traditional approach of developing AI systems fails at addressing this dilemma, either by gathering too much data and hence posing privacy threats or by not collecting sufficient data leading to inaccurate AI systems, federated learning has been developed to partially solve this problem, but this approach also brings in issues of its own.

Limitations of standalone AI – opaqueness, data vulnerability, centralization and privacy threats - are exactly the issues that the architecture of Blockchain aims to solve through its characteristics of transparency and decentralization, laying down the foundation for convergence discussed in the upcoming section.

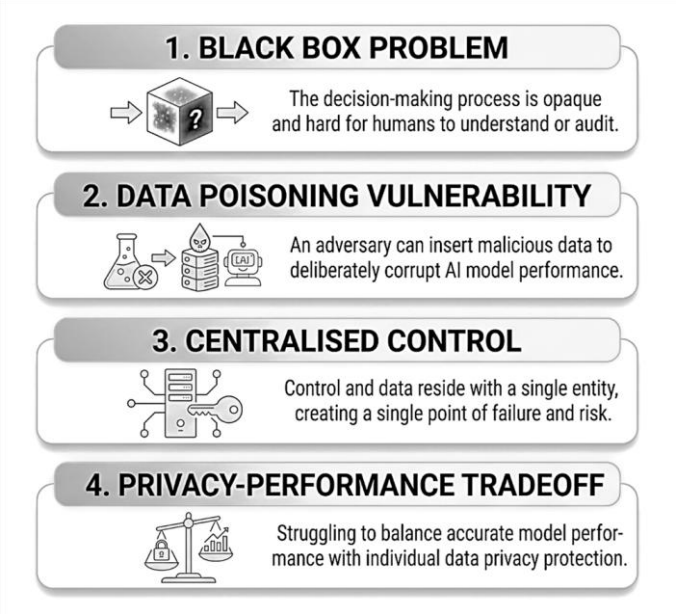


Figure 4: The Four Vulnerabilities of Standalone AI

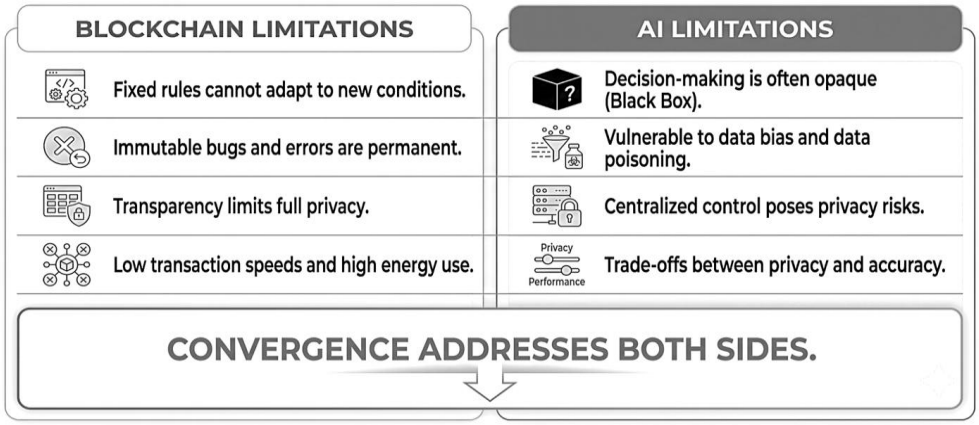


Figure 5: Where Each Technology Falls Short

6.5. BLOCKCHAIN AND AI CONVERGENCE

Although Blockchain technology is highly effective at ensuring a strict level of security and maintaining data immutability during transactions, deploying its infrastructure can be costly and time-consuming. In many scenarios where absolute security is paramount, preventing data tampering is essential. However, certain transactions can still be adequately secured without resorting to such an expensive approach. Conversely, Artificial Intelligence (AI) excels at providing predictions and future event estimations based on historical data. A unified framework that combines the features of Blockchain and AI allows transactions to be processed either adaptively beforehand or verified afterwards (shown in Figures 1 and 2). Data processed through AI can be fed directly into the Blockchain infrastructure; dynamically, the outputs of Blockchain transactions can also serve as inputs for AI models.

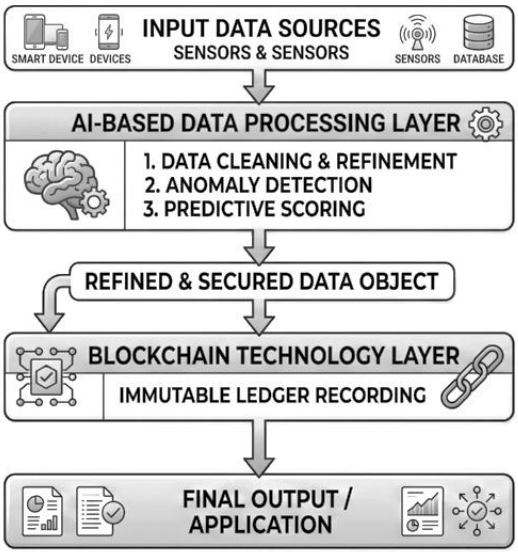


Figure 6: Use of AI before Blockchain

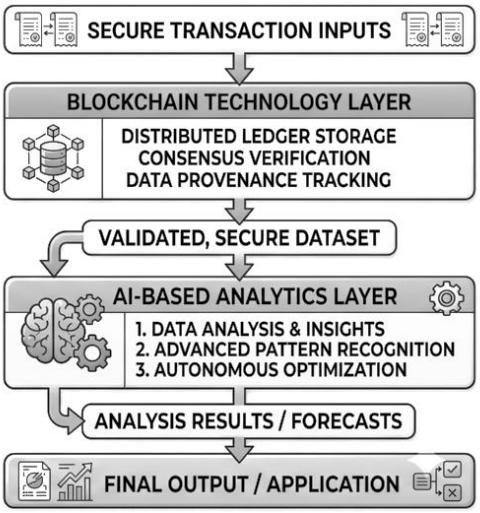


Figure 7: Use of AI after Blockchain technology

Numerous researchers have integrated these two paradigms to leverage their mutual advantages. For instance, Omohundro suggested that combining smart contracts with AI systems can yield significant societal benefits. He talked about the problems with smart contracts. They are good at handling transactions, but they struggle when dealing with the physical world or changing conditions. To fix these issues, an extra layer of intelligence is added to the system. This layer uses an AI model before the Blockchain ledger. It helps make decisions by understanding the context before a transaction is recorded forever. Adding this single layer solved the problems with handling real-life situations and changing conditions. The author also said that AI is important for converting data from sensor devices into a format that smart contracts can understand. This conversion helps smart contracts understand the world. In a system that combines Blockchain and AI, smart contracts can do more than digital transactions. They can interact with the world. For example, they can move objects. Control them in real-life conditions. Smart contracts and AI work together to make this happen. The AI helps smart contracts understand data from the world. This way smart contracts can make decisions. Also, authors Zyskind et al. [4] have also experimented on Blockchain technology and developed a combined framework that consists of Blockchain and AI, which is used for databases to promote increased sharing of personal data by individuals. People use Traditional Blockchain to share their information with others. When we bring everything together, the things that make databases work are really important, for teaching Artificial Intelligence how to make guesses, suggestions or choices. Traditional Blockchain and databases work together to help Artificial Intelligence do its job better. The AdaptChain model is something that was talked about before. It is a model that brings together Reinforcement Learning and Blockchain technology. This new model helps Artificial Intelligence and Blockchain technology work together. Artificial Intelligence systems have some problems. They are not always fair. You do not know what is going on. Blockchain networks have their set of problems, like being too rigid, using too much energy and not being able to handle a lot of people. The AdaptChain model solves all these problems. It helps people trust each other and makes sure everyone is responsible, for what they do. The AdaptChain model also makes sure everything is clear and easy to understand.

AdaptChain is a model that brings a lot of things together. It uses something called reinforcement learning to adjust the consensus parameters as needed. The AdaptChain model has two parts: the state space and the action space. The state space in the AdaptChain model looks at things like how long it takes for transactions to happen how long it takes for blocks to spread how much work the AdaptChain model can handle and how much energy it uses. The action space in the AdaptChain model includes things like changing the size of blocks how often

leaders are chosen, and how many people are on a committee. The reward function in the AdaptChain model tries to find a balance between how work the Blockchain can handle how long things take how much energy it uses and being fair. This balance is what helps make the Blockchain work better. The AdaptChain model is about making the Blockchain work better. The reinforcement learning model is trained using synthetic transaction loads to establish baseline policies, where rewards are continuously computed from live performance metrics (throughput, latency, energy) and maintain stability on invariants such as threshholds, maximum block size, etc.

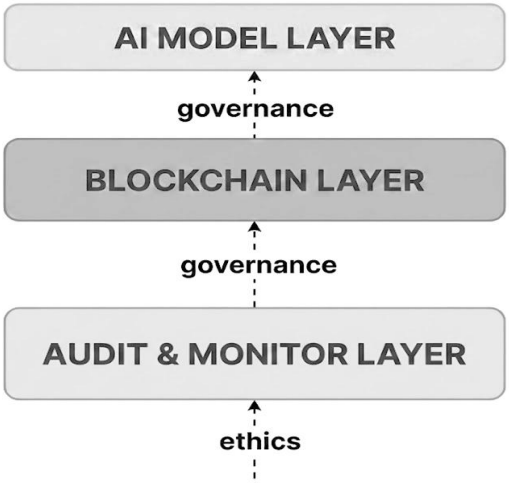


Figure 8: AdaptChain framework

Other than this, AdaptChain also ensures basic ethical issues by incorporating an additional layer in the combined model (shown in Figure 3). Here the model protects contributions. Differential privacy noise is added before updates leave devices. This way sensitive gradients are shielded. Lightweight homomorphic encryption is used during aggregation for protection. The model uses a DAO mechanism. This mechanism helps enforce issues. Smart contracts are used to monitor fairness metrics. They trigger audits when certain thresholds are violated. Stakeholders vote using on-chain ballots. Quorum rules are in place. These rules ensure that decisions are collective. This approach helps avoid bias, from individuals. Disputes are handled through a contract. The outcomes of disputes directly affect AI models and consensus rules.

This chapter has connected two technologies: Blockchain and AI. The chapter has explained how Blockchain's journey from a trustless payment system to a foundation for supply chains, healthcare records, smart contracts, and digital identity — demonstrating its strength in immutability, decentralisation, and transparency. What emerges from placing these two technologies side by side is not a competition but a striking complementarity. Blockchain's verifiable, tamper-proof infrastructure offers exactly what AI lacks in accountability and trust, while AI's adaptive intelligence offers exactly what Blockchain lacks in cognition and responsiveness. This realisation is materialised in converged models like AdaptChain, forms the conceptual foundation.

PART – II:

ARCHITECTURES, GOVERNANCE, AND SECURITY

CHAPTER 7

ARCHITECTURE OF DECENTRALIZED ARTIFICIAL INTELLIGENCE

Contributed by: Pradip Sahoo, Arunima Sinha, Subhadeep Guin, Dr. Payal Bose

7.1 INTRODUCTION

Artificial intelligence systems of the past decade have, almost without exception, been built on a centralized premise: a single organization collects data, trains a model on infrastructure it controls, and exposes that model to users through an interface it also controls. This premise has produced remarkable capability, but it has also concentrated an unusual amount of power, risk, and liability in a small number of institutions. Centralized AI systems suffer from at least three structural weaknesses that motivate the subject of this chapter. First, they create single points of failure: a breach, an outage, or a change of policy at the central node propagates instantly to every dependent user. Second, they obscure data provenance and usage, which erodes the trust of the individuals and organizations whose data trains the model. Third, they concentrate governance — decisions about what the model is permitted to do, who may use it, and how disputes are resolved — in the hands of whoever operates the server, with little recourse for affected parties. Decentralized Artificial Intelligence (DAI) is the name given to a family of architectures that attempt to address these weaknesses by distributing the functions of data custody, model training, verification, and governance across a network of participants, using blockchain and related distributed ledger technologies (DLTs) as the coordinating substrate. The premise is straightforward to state: if no single party controls the data, the model, or the rules, then no single party can unilaterally compromise, corrupt, or capture the system. The architectural reality is considerably more complicated, and unpacking that complexity is the task of this chapter. It is useful, before proceeding further, to be explicit about what DAI is, how it is constructed, why it has attracted sustained research and commercial interest, and from where its constituent techniques originate.

7.1.1 What: Defining the Object of Study

Decentralized Artificial Intelligence refers to the design, training, validation, and governance of machine learning systems using decentralized infrastructure — principally blockchain ledgers and smart contracts — in place of a centrally administered server. The term should not be read as a single technology but as an architectural pattern that recurs, with variation, across federated learning frameworks, multi-agent systems, token-based AI marketplaces, and on-chain governance structures. What unifies these otherwise diverse implementations is a

common architectural commitment: data remains local to the node that produced it, computation is distributed across many participants rather than concentrated in one, and the rules governing coordination, incentives, and dispute resolution are encoded in tamper-evident, auditable form rather than left to the discretion of an operator. This definition deliberately separates DAI from two adjacent but distinct concepts with which it is frequently conflated. It is not merely federated learning, which can be — and often is — orchestrated by a single coordinating server without any blockchain component at all. Nor is it merely "AI on a blockchain" in the narrow sense of running inference inside a smart contract, which remains computationally impractical for anything beyond trivial models. Rather, DAI is the architectural synthesis of distributed learning techniques with blockchain-based coordination, verification, and governance — a synthesis whose component layers are examined in detail in Section 7.3.

7.1.2. The Mechanics of Decentralization

The mechanics by which DAI systems achieve decentralization vary across implementations, but a recurring pattern is evident across the literature. Edge devices or institutional nodes train models locally on data that never leaves their custody. Periodically, these nodes submit model updates — gradients, weights, or aggregated parameters rather than raw data — to a coordination layer. That coordination layer, typically implemented through one or more smart contracts, verifies the authenticity and plausibility of each submission, aggregates accepted updates into a new global model, and records the transaction on an immutable ledger. A governance layer, which may take the form of a Decentralized Autonomous Organization (DAO), reputation system, or hybrid arrangement, determines who may participate, how disputes over model quality or fraud are adjudicated, and how participants are rewarded or penalized for their contributions. Two implementation details recur with sufficient frequency to merit early mention because they shape nearly every architectural decision examined later in this chapter. The first is the on-chain/off-chain division of labor, because blockchain consensus is computationally expensive and slow relative to centralized computation, virtually every practical DAI system performs model training off-chain and reserves the blockchain for coordination, verification, and the recording of state transitions. The second is the substitution of model parameters for raw data as the unit of exchange — a substitution that mitigates, but does not eliminate, privacy risk, since model updates themselves can leak information about the underlying training data through inference and model-inversion attacks.

7.1.3. The Motivating Rationale

The rationale for pursuing decentralized architectures rests on three pillars that appear consistently across the literature surveyed for this chapter: trust, resilience, and accountability.

Trust: Centralized platforms ask users to trust an operator's unverifiable claims about how data is used and how decisions are made. Blockchain's append-only ledger replaces that unverifiable claim with a cryptographically verifiable record, at least for the subset of system behavior that is recorded on-chain.

Resilience: A system with no single coordinating server has no single point at which an outage, a denial-of-service attack, or a unilateral policy change can disable the entire network. Distributing computation and storage across many nodes — exemplified by hybrid storage architectures combining content-addressable systems such as IPFS with archival networks such

as Filecoin — improves fault tolerance, though, as Section 7.5 will argue, it does not eliminate systemic risk so much as relocate it.

Accountability: Smart contracts encode the rules of engagement in advance and execute them automatically, which in principle removes the discretion that allows a centralized operator to apply rules selectively or retroactively. Empirical work on milestone-based fund release in crowdfunding platforms, for instance, demonstrates measurable reductions in fund misappropriation when disbursement is conditioned on independently verified progress rather than left to a platform's unilateral judgment.

These motivations are genuine, and the empirical evidence reviewed in this chapter shows that decentralized architectures can deliver measurable improvements on specific metrics — fraud detection latency, transaction cost, and perceived trustworthiness among them. The chapter's analytical contribution, however, lies in showing that each of these gains is purchased at a specific architectural cost, and that the costs are not evenly distributed across the layers of the system that delivers the gain.

7.1.4. Intellectual and Technical Origins

DAI is not a singular invention but a convergence of three research trajectories that developed largely independently before being drawn together over the past decade. The first trajectory is distributed ledger technology itself, originating in Nakamoto's design for Bitcoin and extended by Ethereum's introduction of programmable smart contracts, which supplied the cryptographic and consensus machinery — hashing, digital signatures, and Proof-of-Work or its successors — on which later DAI systems would depend. The second trajectory is privacy-preserving distributed machine learning, principally federated learning, which emerged from the recognition that centralizing sensitive data (medical records, financial transactions, on-device behavioral data) for model training was both a regulatory liability and a security risk; federated learning's central insight — that only model updates, not raw data, need traverse the network — became the basis on which blockchain-coordinated DAI systems would later build. The third trajectory is the theory and practice of decentralized governance, drawing on mechanism design and on the early experiments of Decentralized Autonomous Organizations, which supplied the vocabulary and tooling — token-weighted voting, staking, slashing — for coordinating the behavior of mutually distrusting participants without a central authority. The remainder of this chapter is organized as follows. Section 7.2 situates DAI architecture within the conceptual landscape of blockchain-enabled systems more broadly, distinguishing the architectural commitments that are specific to AI workloads from those inherited wholesale from general-purpose blockchain design. Section 7.3 presents a layered reference architecture, synthesizing patterns observed across the reviewed literature and worked examples. Section 7.4 examines two contrasting empirical cases — a fraud-resistant crowdfunding platform and an early miner-selection model — to ground the architecture in implemented systems. Section 7.5 offers a critical analysis of the trade-offs the architecture imposes. Section 7.6 surveys contemporary relevance and emerging directions, and Section 7.7 concludes.

7.2. BLOCKCHAIN AS THE COORDINATING SUBSTRATE OF DECENTRALIZED AI

Although this chapter is concerned with the architecture of decentralized AI broadly construed, blockchain technology occupies a structurally privileged position within that architecture: it is the layer that makes verifiable decentralization possible in the first place. It is therefore necessary to examine, with some care, what blockchain specifically contributes to a DAI system, and to distinguish those contributions from properties that are sometimes attributed to blockchain but in fact belong to adjacent technologies such as peer-to-peer storage or cryptographic privacy protocols.

7.2.1 Foundational Properties of Blockchain Relevant to AI Coordination

A blockchain is, at its core, a distributed and append-only ledger maintained collectively by a network of nodes, each holding a synchronized copy of the same transaction history. Three properties of this structure are architecturally relevant to AI systems specifically. Immutability follows from the chaining of blocks by cryptographic hash: each block contains the hash of its predecessor, so that altering any historical block invalidates every subsequent hash and is therefore detectable by any participant. For a DAI system, immutability means that a record of which model update was submitted by which node, at which time, and with what verification outcome, cannot be retroactively altered by any single party — including the operator of the coordination infrastructure. This is the property that underwrites the audit trails described as a core advantage of blockchain-enabled crowdfunding and fraud-detection systems. Consensus mechanisms — historically Proof-of-Work, and increasingly Proof-of-Stake and Byzantine-fault-tolerant variants — solve the problem of how a network of mutually distrusting nodes agrees on a single, canonical version of the ledger without appealing to a central arbiter. For DAI specifically, the consensus layer is what allows competing model updates, submitted by different participants in the same training round, to be ordered, validated, and reconciled into a single agreed-upon global model state. Programmability, introduced at scale by Ethereum's smart contracts, allows the rules governing participation, validation, and reward to be expressed as executable code deployed on the ledger itself, rather than as policy enforced by a human administrator. This is arguably the single most consequential property for DAI architecture, because it is what allows the coordination function previously performed by a centralized training server.

7.2.2 What Blockchain Does Not, by Itself, Provide

A precise architectural account requires equal attention to what blockchain does not supply, because a great deal of imprecision in both popular and academic discourse arises from attributing to “blockchain” capabilities that are, in fact, contributed by separate components bolted onto it.

Blockchain does not provide privacy - A public ledger is, by design, visible to every participant; absent additional cryptographic machinery such as zero-knowledge proofs, differential privacy, or secure multiparty computation, every transaction — including the metadata surrounding a model update — is exposed to network-wide observation. Privacy in DAI systems is therefore the contribution of Layer 2 in the architecture developed in Section 7.3, not of the ledger itself.

Blockchain does not provide scalable storage- Storing large model artifacts, training datasets, or media files directly on-chain is prohibitively expensive given the cost of writing to consensus-secured storage; practical systems instead anchor a content hash on-chain while storing the artifact itself off-chain, typically using systems such as IPFS, OrbitDB, or Filecoin.

Blockchain does not, by itself, perform machine learning- The computational demands of model training make on-chain training infeasible at any meaningful scale; the learning itself occurs off-chain, typically at the edge, with the blockchain reserved for coordination and verification.

7.3. A LAYERED REFERENCE ARCHITECTURE FOR DECENTRALIZED AI

Synthesizing the architectural patterns observed across contemporary decentralized AI systems, this section proposes a five-layer reference architecture for decentralized AI. The architecture is presented as a synthesis rather than a reproduction of any single implementation: no existing system organizes its components in exactly five layers, but the functions commonly performed in decentralized AI environments decompose naturally into the five strata depicted in Figure 7.1. The purpose of presenting the architecture in layered form is analytical rather than merely descriptive — it allows the trade-offs examined in Section 7.5 to be attributed to specific architectural layers rather than treated as undifferentiated properties of decentralized artificial intelligence in the abstract.

Each layer is examined below in turn, with attention to the specific architectural function it performs, the technologies most commonly used to implement it, and the dependency it creates on the layer beneath it.

7.3.1 Layer 1: The Distributed Ledger Layer

The foundation of the architecture is the distributed ledger itself, which supplies the immutable transaction record, the consensus mechanism, and the cryptographic hashing on which every higher layer ultimately relies. Ethereum remains one of the most widely adopted platforms for this layer because of its mature smart contract ecosystem; Hyperledger appears disproportionately in permissioned, organization-bounded contexts such as healthcare consortia, where the parties are known in advance and public, permissionless consensus is unnecessary overhead; Polkadot and Cosmos appear in designs oriented toward cross-chain interoperability. The architectural significance of this layer is best understood by what it makes possible for the layers above it: a tamper-evident record that any participant can independently verify without trusting any other participant.

This is also the layer at which one of the most significant limitations of decentralized AI systems originates. The computational overhead of achieving distributed consensus imposes a throughput ceiling and a latency floor that no higher layer can architect away. The reliability of decentralized consensus systems is commonly expressed using the Byzantine fault tolerance condition. A distributed network can tolerate malicious or faulty participants only if the total number of nodes satisfies the following relationship:

$$n \geq 3f + 1 \qquad (1)$$

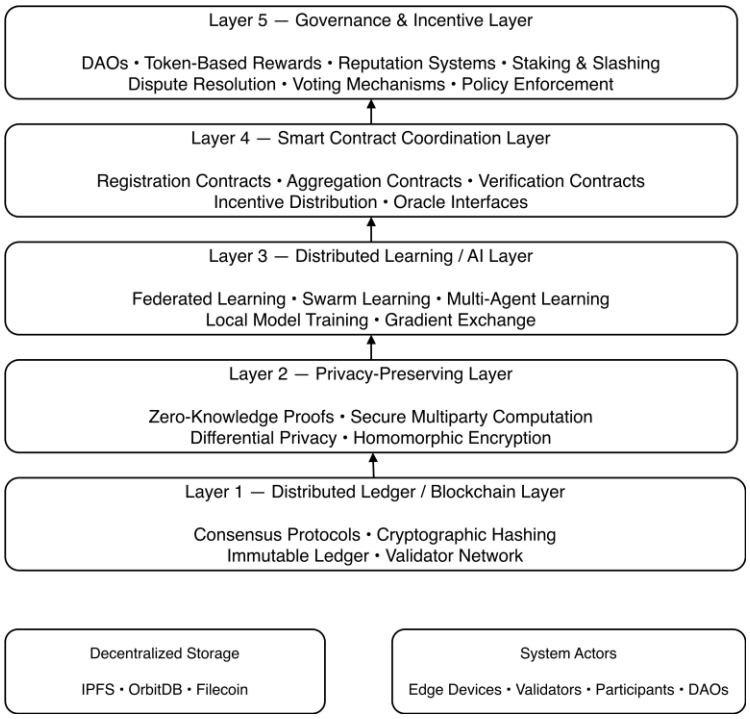


Figure 1. Layered reference architecture of decentralized AI systems.

where: n = total participating nodes, f = maximum number of faulty nodes. This relationship indicates that a consensus network requires at least three correctly functioning nodes for every potentially faulty participant in order to preserve consensus integrity. The latency introduced by distributed consensus can be represented as the sum of network communication, validation, and block generation delays:

$$T_{consensus} = T_n + T_v + T_b \qquad (2)$$

where: T_n= network propagation delay; T_v= transaction validation; T_b= block creation delay.

One implemented hybrid blockchain–AI system illustrates this concretely: its measured transaction throughput of 47.3 transactions per second is roughly a third of the throughput achieved by two centralized comparison platforms (Table 7.1), a gap attributable specifically to the consensus overhead of the distributed ledger layer rather than to any deficiency in the layers built atop it.

7.3.2 Layer 2: The Privacy-Preserving and Cryptographic Layer

Because a distributed ledger is, by default, transparent to every participant, a second layer is required to reconcile that transparency with the confidentiality demands of AI workloads — particularly in domains such as healthcare and finance, where the data underlying model training is subject to regulatory protection. Four principal techniques commonly operate at this layer: zero-knowledge proofs, which allow one party to prove that a model update is valid without revealing the underlying data; secure multiparty computation, which allows several parties to jointly compute a function over their combined inputs without any party learning

another's input; differential privacy, which injects calibrated statistical noise into shared model updates; and homomorphic encryption, which permits computation directly on encrypted data.

Differential privacy mechanisms are commonly controlled by a privacy budget parameter, denoted by ε, which governs the trade-off between privacy protection and model utility.

where: $\varepsilon = privacybudget$

Smaller values of ε provide stronger privacy guarantees by introducing greater uncertainty into the shared updates, whereas larger values preserve model accuracy at the expense of weaker privacy protection. Differential privacy mechanisms typically inject carefully calibrated noise into shared model parameters. The process may be represented as:

$$M(x) = f(x) + Lap\left(\frac{\Delta f}{\epsilon}\right) \quad (3)$$

where: f(x) = original function; Δf = sensitivity; ϵ = privacy budget

Smaller values of ε provide stronger privacy protection but may reduce model accuracy, creating an explicit trade-off between privacy and utility. This layer exists specifically to mitigate a vulnerability introduced by Layer 3: because federated learning exchanges model parameters rather than raw data, it is tempting to assume that the privacy problem is solved by the architecture of Layer 3 alone. However, this assumption is unsafe — gradient and parameter updates remain susceptible to model-inversion and membership-inference attacks that can reconstruct properties of the original training data. Layer 2 is therefore not an optional hardening measure but a structurally necessary complement to Layer 3, and a DAI architecture that omits it should be understood as offering a materially weaker privacy guarantee than its "federated" branding might suggest.

7.3.3 Layer 3: The Distributed Learning Layer

This is the layer most readers will associate with "AI" in the conventional sense: the actual training of machine learning models. Federated learning is the dominant paradigm and serves as the primary learning protocol in many decentralized AI systems, with swarm learning and multi-agent reinforcement learning appearing in specialized contexts such as collaborative hospital networks and autonomous vehicle coordination. The architectural commitment shared across these variants is that training occurs locally — on the edge device, the institutional server, or the individual node — and only the resulting model parameters, not the raw data, are transmitted onward to Layer 4 for aggregation.

The aggregation of local model updates is commonly performed using the Federated Averaging algorithm:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k \quad (4)$$

where: w_k = local model parameters; n_k = local training samples; n = total samples. The equation assigns greater influence to nodes contributing larger training datasets. It is worth pausing on a frequently underexamined architectural consequence of this layer's design: data heterogeneity. Because each participating node trains on its own local data distribution, which

may differ substantially from the distributions held by other nodes (a hospital in one region may see a different patient population than a hospital in another), the aggregated global model must reconcile updates derived from non-identically distributed data. The impact of heterogeneous local datasets on the global model may be represented through the aggregation error:

$$E_g = \sum_{k=1}^{K} p_k E_k \quad (5)$$

where: E_g = global model error; E_k = local model error; p_k = contribution weight

This formulation highlights that the quality of the global model depends not only on the accuracy of individual local models but also on the distribution of training data across participating nodes. This is a substantive machine learning challenge in its own right, and it is compounded — not resolved — by the decentralization of the surrounding architecture, since no central party has visibility into the full set of local distributions against which to calibrate the aggregation. The communication overhead associated with decentralized learning may be approximated as:

$$C = N \times S \times R \quad (6)$$

where: N = participating nodes; S = update size; R = communication rounds. The communication cost grows linearly with both the number of participants and the frequency of model synchronization.

7.3.4 Layer 4: The Smart Contract Coordination Layer

If Layer 1 supplies the substrate of verifiable trust, Layer 4 is where that trust is operationalized into the specific coordination functions a centralized AI platform would otherwise perform through a trusted server. Four recurring contract types commonly appear within this layer. Registration contracts onboard new participants, typically requiring a security deposit or verifiable identity credential as a precondition of participation. Aggregation contracts merge accepted local model updates into a new global model, replacing the function of a centralized parameter server. Verification contracts authenticate submitted updates and screen for poisoned or low-quality contributions, often using a challenge-response protocol against held-out synthetic data. Governance contracts encode the rules for access control, incentive distribution, and the escalation of disputes. The effectiveness of verification mechanisms may be approximated using the verification probability:

$$P_v = \frac{V_a}{V_t} \quad (7)$$

where: V_a = accepted updates; V_t = total submitted updates

Higher values of P_v indicate that a larger proportion of submitted updates satisfy the validation criteria imposed by the smart contract, thereby improving the reliability of the aggregation process. The milestone-gated coordination logic typical of this layer — in which funds, or by direct analogy, model updates, are released or accepted only once a quorum of independent verifiers has attested to a precondition being met — is well illustrated by the conditional fund-release mechanism employed in blockchain-enabled crowdfunding architectures. The same

conditional-release logic that governs milestone-based fund disbursement is structurally identical to the logic governing milestone-based model aggregation in a DAI training pipeline; both substitute a verifiable, automatically enforced threshold for the discretionary judgment of a centralized operator.

7.3.5 Layer 5: The Governance and Incentive Layer

The topmost layer governs the behavior of the network's human and institutional participants rather than its data or models directly. Four incentive mechanisms commonly appear within this layer: token-based rewards, which compensate participants in proportion to the accuracy and timeliness of their contributions; reputation systems, which track historical reliability and weight future influence accordingly; staking and slashing, which require participants to post collateral that is forfeited upon detected dishonesty; and hybrid arrangements combining the three. The economic penalty associated with dishonest behavior may be represented as:

$$Penalty_i = S_i P \tag{8}$$

where: S_i = staked amount; p = penalty factor. Larger stakes therefore expose dishonest participants to greater financial losses, creating an economic incentive for honest participation.

The allocation of incentives may be expressed as:

$$Reward_i = \frac{C_i}{\sum C_i} R \tag{9}$$

where: C_i = contribution; R = reward pool. This formulation allocates incentives in proportion to the relative contribution of individual participants, thereby encouraging active participation and discouraging free-riding behavior.

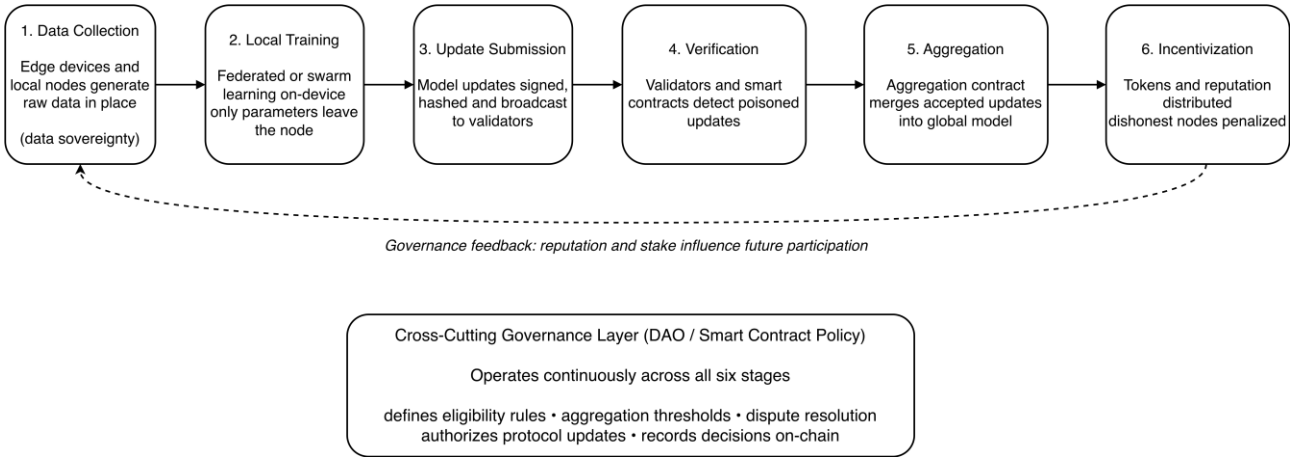


Figure 2. Operational lifecycle of a blockchain-coordinated decentralized AI system

Figure 2 traces this coordination function across the full operational lifecycle of a DAI system, from local data collection through incentivization, with a feedback arrow indicating that governance outcomes (Stage 6) condition the eligibility of participants in subsequent rounds (Stage 1) — a closed loop that is one of the architecture's principal advantages over static, one-

shot centralized systems, but also, as Section 7.5 will argue, one of its principal sources of long-run governance risk.

Although these mechanisms appear conceptually distinct, practical governance systems frequently combine them to estimate the trustworthiness of participating nodes. A participant's trust level may therefore be represented as a weighted combination of reputation, historical reliability, and economic commitment:

$$Trust_i = \alpha R_i + \beta H_i + \gamma S_i \quad (10)$$

where: R_i = reputation score; H_i = historical reliability; S_i = staking contribution; α, β, γ = weighting coefficients. This formulation allows governance systems to balance reputation, historical behavior, and economic commitment. Participant reputation may evolve over time according to observed behavior:

$$Rep_{t+1} = Rep_t + \Delta Rep \quad (11)$$

where: Rep_t = current reputation; ΔRep = reputation change. Positive contributions increase the participant's reputation, whereas dishonest or low-quality behavior reduces future influence within the governance process. In practical decentralized environments, trust is not necessarily permanent and may decline over time in the absence of continued participation or positive contributions. This behavior may be represented using a trust decay model:

$$Trust_t = Trust_0 e^{-\lambda t} \quad (12)$$

where: $Trust_0$ = initial trust score; t = elapsed time; λ = decay constant. The model reflects the assumption that inactive or unverified participants gradually lose influence within the governance process, thereby encouraging continued engagement and honest participation. Decentralized Autonomous Organizations represent the most architecturally ambitious implementation of this layer, encoding governance rules — who may vote, what threshold constitutes a passed proposal, how disputed model contributions are adjudicated — directly into smart contracts rather than leaving them to an administrator's discretion. Figure 7.3 positions the three governance models — centralized, hybrid, and DAO-based — along a spectrum of decentralization, and previews an argument developed at length in Section 7.5: that no point on this spectrum eliminates the dependency on trust so much as relocates it, in the DAO case toward whichever participants hold the largest token stake or exert the greatest influence over validator selection.

7.3.6 Cross-Cutting Concerns: Decentralized Storage and System Actors

Two further components do not map cleanly onto a single layer of the architecture because they are accessed by, and serve, multiple layers simultaneously. Decentralized storage systems — IPFS for content-addressable media, OrbitDB for frequently updated records such as comments and live progress reports, and Filecoin for long-term archival persistence — provide the off-chain capacity that Layer 1 cannot economically provide, while remaining anchored to Layer 1 through stored content hashes. System actors — edge devices, validators, ordinary participants, and DAOs — populate every layer of the architecture simultaneously, occupying different roles

depending on the function being performed at a given moment: a single institutional node may be a data source at Layer 3, a verifier at Layer 4, and a voting member of a DAO at Layer 5.

Table 1 summarizes the five layers, the dominant technologies implementing each, and the specific vulnerability each layer is designed to mitigate — a synthesis that makes explicit the dependency chain examined critically in Section 7.5.

Table 1. Layered architecture of decentralized AI systems, synthesizing functions, technologies, and mitigated vulnerabilities.

Layer	Primary Function	Representative Technologies	Vulnerability Mitigated
1. Distributed Ledger	Establishes immutable, consensus-verified transaction history	Ethereum, Hyperledger, Polkadot, Cosmos	Centralized record tampering; single point of administrative control
2. Privacy / Cryptographic	Reconciles ledger transparency with data confidentiality	Zero-knowledge proofs, SMPC, differential privacy, homomorphic encryption	Inference and model-inversion attacks against shared parameters
3. Distributed Learning	Trains models locally without centralizing raw data	Federated learning, swarm learning, multi-agent reinforcement learning	Centralized data custody; single point of data-breach failure
4. Smart Contract Coordination	Automates aggregation, verification, and incentive logic	Registration, aggregation, verification, and governance contracts	Discretionary, unaccountable platform-operator decision-making
5. Governance & Incentive	Aligns participant behavior with collective network objectives	DAOs, token rewards, reputation scores, staking/slashing	Free-riding, malicious participation, and unaccountable rule-setting

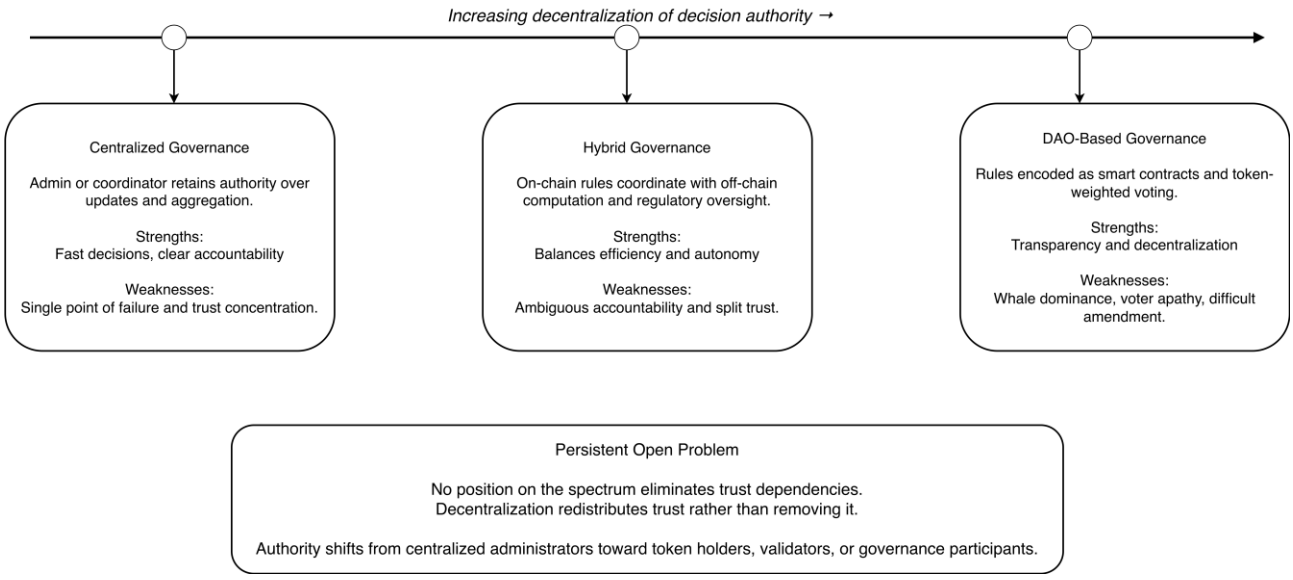


Figure 3. Governance models in decentralized AI, positioned along a centralization–decentralization spectrum.

7.4 CASE STUDIES IN IMPLEMENTED ARCHITECTURE

The reference architecture developed in Section 7.3 is a synthesis abstracted from the literature; this section grounds that synthesis by examining two implemented or prototyped systems in detail, chosen specifically because they sit at different points in the architecture's maturation and emphasize different layers. The first, BlockFund, is a fully specified hybrid blockchain-AI system for crowdfunding, emphasizing Layers 3 and 4. The second, the Decentralized AI-enabled Blockchain Network (DAIBCN), is an earlier proof-of-concept that emphasizes Layer 1 and the AI-assisted optimization of consensus itself — a reminder that the relationship between AI and blockchain in DAI architecture runs in both directions: blockchain coordinates AI, but AI can also be used to optimize blockchain.

7.4.1 BlockFund: A Hybrid Blockchain–AI Architecture for Crowdfunding

BlockFund is a hybrid blockchain–AI crowdfunding platform that combines an Ethereum-based smart contract suite with a multimodal AI fraud-detection ensemble. The system is organized around four functional components: a blockchain layer comprising a Platform Contract, Campaign Factory Contract, Campaign Contract, and Governance Contract; an AI layer combining transformer-based text analysis, graph neural network transaction analysis, LSTM-based time-series analysis, and a gradient-boosted creator-reputation model, unified by a meta-learner; a decentralized storage layer using IPFS, OrbitDB, and Filecoin; and a user interface layer supporting a range of wallet integration types. The architectural feature most relevant to this chapter's analytical focus is BlockFund's milestone-based fund release mechanism, which substitutes staged, independently verified disbursement for the conventional all-or-nothing crowdfunding model. Funds are held in escrow within the Campaign Contract and released only once a quorum of authorized verifiers attests that a milestone has been met — the conditional-release logic depicted abstractly in Figure 7.2 above. This is a direct architectural instantiation of the principle, articulated in Section 7.1.3, that smart contracts can substitute automatically enforced, pre-committed rules for the discretionary judgment of a centralized platform operator. The reported performance results merit close reading rather than uncritical interpretation, because they illustrate the architecture's trade-offs concretely rather than abstractly. BlockFund's measured transaction throughput of 47.3 transactions per second is substantially lower than the 124.6 and 98.2 transactions per second achieved by two centralized comparison platforms, a gap attributable to blockchain consensus overhead — precisely the Layer 1 limitation identified in Section 7.3.1. Against this throughput cost, the system demonstrates a transaction cost of $0.42, roughly half that of the lower-cost centralized comparator, and a fraud-detection true-positive count of 34 out of a manually verified set, against 22 for a rule-based comparator and 30 for manual review, with a mean detection time of 3.2 hours against 27.4 and 48.7 hours, respectively.

Two methodological caveats are warranted before these findings are generalized. First, the performance comparison was conducted on a simulated workload of 120 campaigns rather than on production traffic, which is an appropriate and common evaluation strategy for a novel architecture but constrains the confidence with which throughput and cost figures can be extrapolated to larger-scale deployment. Second, the fraud-detection dataset was deliberately

constructed to contain both legitimate and adversarial campaign behavior in a controlled ratio. This is methodologically sound for evaluating detection sensitivity, but it means the reported true-positive and false-negative rates describe performance against a known, curated fraud distribution rather than against the open-ended and evolving fraud tactics a deployed system would eventually face. The system therefore illustrates the strengths of the proposed architecture while also highlighting the challenges associated with generalizing experimental results to real-world environments.

7.4.2 DAIBCN: AI-Optimized Consensus as the Inverse Architectural Pattern

DAIBCN represents an earlier and architecturally distinct proof-of-concept: a Decentralized AI-enabled Blockchain Network in which artificial intelligence is used not to ride atop a blockchain coordination layer, as in BlockFund, but to optimize the blockchain's own consensus mechanism. The system replaces the computationally expensive Proof-of-Work puzzle-solving competition among miners with a decision-tree-based miner-selection algorithm that scores candidate nodes against attributes including hardware type, CPU speed, bandwidth, historical reward points, and a calculated trust level. This approach is worth foregrounding precisely because it inverts the directionality implicit in much contemporary DAI discourse, where "decentralized AI" is typically read as "AI coordinated by blockchain." DAIBCN instead instantiates "blockchain optimized by AI" — a reminder that the relationship between the two technologies in this architectural space is genuinely bidirectional, and that a complete account of DAI architecture should not treat blockchain as a fixed, AI-agnostic substrate. The system demonstrates substantial reductions in per-transaction energy consumption and block-commit latency relative to a traditional Proof-of-Work baseline, largely due to the elimination of redundant computational competition among non-selected miners.

The total computational energy consumed by participating nodes may be expressed as:

$$E_{total} = \sum_{i=1}^{N} E_i \qquad (13)$$

where: E_i = energy consumed by node I; N = total participating nodes. Reducing the number of active consensus participants decreases the cumulative energy requirement of the network, which explains the improved efficiency observed in AI-assisted consensus mechanisms. The architectural significance of this case for the present chapter is twofold. First, it demonstrates that the boundary between Layer 1 (distributed ledger) and Layer 3 (distributed learning) in the reference architecture of Figure 7.1 is not as clean in practice as the layered diagram implies: here, machine learning logic is embedded directly within the consensus layer itself, rather than sitting atop it as a separate workload. Second, and more cautionarily, the proposed decision-tree selection criteria — reward points, trust level, historical reliability — are themselves attributes that must be tracked, updated, and protected against manipulation, which means the architecture has not eliminated the governance problem examined in Section 7.3.5 so much as pushed it one layer further down, into the selection logic that determines who gets to mine in the first place. A participant capable of inflating its own trust score or reward-point history would gain disproportionate influence over block production — an early illustration of the general trust-relocation argument developed at length in Section 7.5.2.

7.4.3 Comparative Reading of the Two Cases

Read together, BlockFund and DAIBCN illustrate that “decentralized AI architecture” is not a single design pattern with interchangeable instances, but a design space within which systems can be positioned quite differently depending on which layer of Figure 7.1 is treated as the primary site of innovation. BlockFund's contribution is concentrated at Layers 3 and 4 — a sophisticated multimodal AI ensemble coordinated through milestone-conditioned smart contracts — while its Layer 1 is comparatively conventional Ethereum infrastructure, inheriting that platform's well-documented throughput limitations. DAIBCN's contribution is concentrated at Layer 1 itself, using a lightweight AI selection heuristic to reduce the computational cost of consensus, while leaving the higher-layer AI workload essentially unaddressed. This comparative reading supports a claim that recurs through the remainder of the chapter: architectural maturity in DAI is uneven across layers, and a system's headline performance figures typically reflect engineering investment concentrated in one or two layers rather than uniform improvement across the full stack depicted in Figure 7.1. Evaluating any single DAI system, therefore, requires asking not simply whether it is “decentralized” but which specific layer's function it decentralizes, and at what cost to the layers it leaves comparatively under-engineered.

7.5 CRITICAL ANALYSIS: THE ARCHITECTURAL TRADE-OFFS OF DECENTRALIZATION

The preceding sections have presented decentralized AI architecture descriptively, layer by layer, with reference to its motivating rationale and to two implemented systems. This section turns to critical analysis. The central argument advanced here is that the architecture examined in Section 7.3 does not resolve the structural weaknesses of centralized AI identified in Section 7.1 so much as redistribute them across a different set of components, each of which introduces a distinct and sometimes underappreciated cost. Four such trade-offs are examined: the throughput-versus-trust trade-off inherent to Layer 1; the relocation, rather than elimination, of trust at Layer 5; the persistence of privacy risk despite Layer 2's mitigations; and the emergence of incentive misalignment as a structural rather than incidental property of the architecture.

7.5.1 The Throughput–Trust Trade-off

Every consensus mechanism examined in Section 7.3.1 purchases its tamper-evidence at the cost of throughput, and the magnitude of that cost is large enough to be architecturally, not merely incidentally, significant. Table 7.2 juxtaposes the throughput and cost figures observed for BlockFund against its centralized comparators, set alongside the efficiency gains demonstrated by DAIBCN over conventional Proof-of-Work systems, to illustrate that the trade-off is not uniform across implementations but depends heavily on which specific consensus optimization, if any, a given system adopts.

Table 2. Performance comparison between a hybrid blockchain–AI architecture and two conventional centralized platforms.

Metric	BlockFund (Hybrid)	Conventional Platform A	Conventional Platform B
Transaction throughput (TPS)	47.3	124.6	98.2

Cost per transaction (USD)	$0.42	$0.87	$0.76
Server instances required	12 (distributed)	37	29
System availability (%)	99.97	99.91	99.88

Table 2 should be read with a specific interpretive caution in mind: the throughput figures and the cost figures are not independent of one another, and treating either in isolation produces a misleading picture of the architecture's net value. Transaction throughput is commonly measured as:

$$TPS = \frac{N_t}{T} \tag{14}$$

where: N_t = processed transactions; T = execution time

The TPS metric provides a direct quantitative measure of the scalability of a decentralized system. The lower throughput is the price paid for the lower per-transaction cost and the markedly lower server-instance requirement; BlockFund's twelve distributed nodes achieve a system availability statistically comparable to the conventional platforms' thirty-seven and twenty-nine servers, suggesting that decentralized consensus, even at a throughput cost, may deliver superior resource efficiency per unit of reliability. System availability may be expressed as:

$$Availability = \frac{Uptime}{TotalTime} \tag{15}$$

where: Uptime = operational duration; Total Time = total observation period. High availability indicates that the distributed infrastructure remains operational despite individual node failures. Whether this trade is favorable depends entirely on the workload: a system handling thousands of low-value, latency-sensitive microtransactions per second will find a 47.3 TPS ceiling disqualifying, while a system handling milestone-gated, comparatively infrequent, high-value disbursements — as in crowdfunding — will not.

DAIBCN's contribution should be read as a partial, rather than complete, answer to this trade-off. By replacing competitive Proof-of-Work mining with AI-assisted node selection, the architecture can reduce the redundant computation that depresses throughput in conventional Proof-of-Work systems. But this gain is purchased by narrowing the set of nodes that participate in any given round of consensus to those scoring highest on the selection criteria — which is to say, the throughput gain is achieved by reducing, not preserving, the breadth of decentralized participation in block production. The architecture has not escaped the throughput–trust trade-off; it has shifted the trade along a different axis, exchanging some of the trust benefit of broad miner participation for a throughput benefit, rather than obtaining both simultaneously.

7.5.2 The Relocation, rather than Elimination of Trust

The most consequential argument of this chapter concerns governance. Section 7.1.3 identified accountability as a primary motivation for decentralized architecture, on the premise that encoding rules in smart contracts removes the discretion that allows a centralized operator to

apply rules selectively. This premise is correct as a description of what smart contracts do, but it is incomplete as a description of what decentralized governance achieves overall, because it leaves unaddressed the question of who writes the smart contract, who controls the validator set that executes it, and who holds the token weight that governs its amendment.

One of the most persistent challenges in decentralized governance is the problem of "whale dominance": token-weighted, "one-token-one-vote" governance structures concentrate decision-making power among the largest stakeholders, precisely the concentration of unaccountable authority that decentralization was meant to dissolve. DAOs may also suffer from voter apathy and from a structural rigidity that makes their rules costly and slow to amend even when amendment is clearly warranted. Figure 7.3 depicted this as a spectrum running from centralized to hybrid to DAO-based governance; the critical claim advanced here is that movement along this spectrum changes who holds trust-conferring authority without changing the fact that someone does. A centralized platform operator who behaves badly is, at minimum, identifiable and potentially subject to legal or reputational sanction; a coalition of anonymous large token-holders who behave badly through a sequence of individually defensible votes may be neither.

DAIBCN's miner-selection criteria, examined in Section 7.4.2, illustrate the same dynamic at a different layer: trust level and reward-point history function as a de facto reputation-based gatekeeping mechanism for participation in consensus, which means the architecture has relocated the question "who decides?" from a centralized mining-pool operator to whichever algorithm and historical-data pipeline determines a node's trust score. The chapter's position is therefore not that decentralized governance is inferior to centralized governance, but that the comparison should not be framed as "trust versus no trust" — it should be framed as a comparison between different distributions of trust, each with its own failure modes, costs of auditing, and avenues for redress.

7.5.3 Persistent Privacy Risk Beneath the Federated-Learning Abstraction

Section 7.3.2 established that Layer 2's cryptographic protections exist specifically because Layer 3's exchange of model parameters rather than raw data does not, by itself, guarantee privacy. Gradient leakage, Sybil attacks, and the practical difficulty of implementing secure aggregation via homomorphic encryption or multiparty computation at scale remain open, unresolved problems rather than solved engineering footnotes. This matters for an evaluation of DAI architecture because the marketing and rhetorical framing of federated and decentralized learning frequently implies a privacy guarantee — "your data never leaves your device" — that the underlying cryptographic mechanisms do not yet fully support without the additional, computationally expensive Layer 2 machinery, which many implemented systems, including both case studies examined in Section 7.4, deploy only partially or not at all.

A further complication, not always foregrounded in architectural discussions of decentralized AI systems, is that zero-knowledge proofs and secure multiparty computation impose their own throughput and latency costs on top of those already imposed by Layer 1's consensus mechanism. A system seeking strong privacy guarantees at Layer 2 and strong decentralization guarantees at Layer 1 simultaneously compounds, rather than offsets, the throughput penalties examined in Section 7.5.1. The architecture, in other words, does not offer a free combination

of privacy and decentralization; it offers a three-way trade-off among privacy, decentralization, and performance, and any given implementation occupies a specific, and rarely optimal-on-all-three-axes, point within that trade-off space.

7.5.4 Incentive Misalignment as a Structural Property

Finally, the incentive mechanisms examined in Section 7.3.5 — token rewards, staking, reputation — are not merely imperfect implementations of a sound design; some degree of misalignment appears to be structurally inherent to any incentive scheme operating in this domain. A participant can rationally maximize expected reward by submitting model updates trained on a minimal subset of available data, since doing so reduces computational cost while still satisfying the formal validation criteria a verification contract can practically check.

The economic motivation of participants may be represented using a simple utility model:

$$U_i = R_i - C_i \qquad (15)$$

where: R_i = reward received by participant I; C_i = computational cost incurred by participant i. Participants maximize utility by increasing rewards while minimizing computational expenditure. This creates the incentive misalignment discussed in this section, where rational behavior may not coincide with the objectives of the overall system. Requiring a deposit as a precondition of participation can deter this free-riding behavior, but it simultaneously raises the barrier to entry for legitimate, resource-constrained participants — precisely the constituency that decentralized, low-barrier participation was meant to serve. This is not a problem that better engineering at any single layer can fully resolve, because it arises from the interaction between layers: Layer 4's verification contracts can only check what is computationally and informationally checkable on-chain, Layer 5's incentive design can only reward what Layer 4 can verify, and Layer 3's local training process is, by the architecture's own privacy-preserving design, opaque to outside inspection. The result is a persistent gap between the behavior the architecture intends to reward and the behavior it is actually capable of measuring — a gap that remains an open problem rather than a solved one.

7.6 CONTEMPORARY RELEVANCE AND EMERGING ARCHITECTURAL DIRECTIONS

The architectural trade-offs identified in Section 7.5 are not merely theoretical; they map onto specific, actively contested design questions in the contemporary DAI literature. This section examines four such questions, each corresponding to an open problem identified across the body of reviewed work, with attention to how each bear on the layered architecture developed in Section 7.3.

7.6.1 Scalability: Layer-2 and Modular Architectures

Scalability remains one of the most significant constraints on the large-scale deployment of decentralized AI systems, a concern directly consistent with the throughput limitations examined in Section 7.5.1. The architectural responses currently under investigation cluster around two strategies: shifting computationally intensive work off-chain while anchoring only summary commitments on-chain, and adopting Layer-2 scaling solutions — rollups and

directed-acyclic-graph-based structures — that batch multiple transactions before committing them to the base ledger. Both strategies are, in the vocabulary of this chapter's reference architecture, attempts to relax the throughput cost of Layer 1 without abandoning its trust guarantees; whether they succeed in doing so without quietly reintroducing centralization at the rollup-operator level remains an open question rather than a settled one.

7.6.2 Privacy-Preserving Computation at Scale

A second active research direction concerns whether zero-knowledge proofs and verifiable computation frameworks can be made efficient enough to support real-time model auditing without the prohibitive overhead identified in Section 7.5.3. Progress here would directly relax the three-way trade-off among privacy, decentralization, and performance described in that section, and would represent the single architectural advance most capable of closing the gap between the privacy guarantees DAI systems are popularly understood to offer and the guarantees their current cryptographic implementations actually deliver.

7.6.3 Post-Quantum Cryptographic Resilience

A further direction concerns the integration of post-quantum cryptographic primitives into the Layer 2 protections examined in Section 7.3.2, motivated by the prospect that sufficiently capable quantum computation could eventually undermine the digital signature and hashing schemes on which both Layer 1 and Layer 2 currently depend. This direction is presently more anticipatory than operational, but its inclusion in contemporary discussions of decentralized AI suggests that the architecture's cryptographic foundations are increasingly regarded as provisional rather than permanent.

7.6.4 Standardization and Cross-Chain Interoperability

Finally, the absence of standardized protocols, benchmarks, and model formats across heterogeneous blockchain platforms and AI frameworks complicates both academic comparison and practical interoperability. A DAI system built on Ethereum cannot, without substantial custom middleware, coordinate directly with one built on Hyperledger or Polkadot, which constrains the architecture's promise of open, permissionless participation to whichever platform a given deployment has committed to in advance. Cross-chain coordination protocols capable of preserving the privacy and integrity guarantees of Layers 1 and 2 while bridging across platforms remain a substantially unresolved engineering and standardization challenge. Each of these four directions targets a specific layer or cross-layer interaction identified in the architecture, which is itself a useful diagnostic. This field has most active research questions are not evenly distributed across the five-layer stack but cluster disproportionately. It suggesting the foundational substrate of the architecture, rather than its higher-level AI or governance logic, remains its principal source of unresolved technical risk.

CHAPTER 8

FEDERATED AND DISTRIBUTED LEARNING SYSTEMS

Contributed by: Kousik Dey, Somen Pandey, Rinku Mahata, Amit Ghosh

8.1 INTRODUCTION

Federated Learning (FL) is a recent technique in machine learning infrastructure that enables the development of models across devices and organizations that are located at various places, without exposing any private data. This chapter explains the concepts associated with FL and distributed learning technologies. It begins with motivations concerning data security law compliance and computational differences that motivate various FL methods, and describes types of federated models (cross-device federations and cross-silo federations). This is followed by the exploration of major aggregation techniques (FedAvg and FedProx) and usage of differential privacy and secure multi-party computation to preserve participants privacy. The chapter also investigates factors such as non-IID data sets, communication efficiency and Byzantine fault tolerance, as well as new research areas (personalized FL and split learning). With the help of illustrations, comparisons and empirical research results, the chapter provides readers the necessary knowledge that can be used when developing new federated systems. The rise of edge computing technology has drastically changed the method of generating data. It is believed that most of the data is generated at the network edge, but traditional machine learning procedures suggest collecting this data at the central data centers to be able to train the model. Such arrangements cause conflict between the proper use of data and the issues of privacy, compliance with regulations and communication capabilities. An innovative method to cope with the current problem is the implementation of Federated Learning (FL), which was first introduced by McMahan in the FedAvg algorithm. In contrast to the conventional practice of sending raw data to a unified storage facility, Federated Learning involves providing each unit with training locally and sending only the information about model improvement in the form of updates pertaining to model gradients and weight changes to the global processing center. The information is subsequently processed at the level of the center and becomes the new state of the global model that will be sent to the devices for further training. The definition of Federated Learning is not restricted to protecting privacy. Several regulated sectors like finance, healthcare, and telecommunication have tight regulations like General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or California Consumer Privacy Act (CCPA) which prevent data sharing between organizations. FL makes it

possible for organizations in these sectors to gain from collective data knowledge without breaking these regulations.

8.2 CENTRALIZED VS. FEDERATED VS. DISTRIBUTED LEARNING

Before one explores the aspects and methods involved in federated learning (FL), one must first understand the dissimilarities between this paradigm and those of other machine learning paradigms. Currently, there are three main machine learning paradigms: centralized learning, distributed learning, and federated learning. Even though the principal objective of all three types of learning is to create correct and efficient models for prediction, there are many differences in functioning. In particular, these three paradigms vary in terms of the system of data storage, computational structure, communication means, scalability, and preservation of privacy principles. Centralized learning is the process of collecting and sharing information from several different sources. The models are trained within a centralized data center that uses powerful computers. This approach often leads to a very successful model as it can access all data in one place. However, it also raises security issues related to privacy, storage and communication costs, and potential violations of applicable laws or policies concerning data sharing between organizations. The primary goal of Distributed Learning Techniques is to improve the efficiency of model training by reallocating computing resources across several machines or servers to run operations more quickly. The data can be divided between computing nodes; however, such systems must be located within the same trusted environment or datacenter, with the nodes being directly connected to each other and engaged in communication and synchronization. In contrast to federated learning, this approach calls for the pooling of local models from various devices or businesses to create the global model, while data rests housed in its local premises. Exchange of sensitive information occurs solely among model and central server for the purposes of protection of data. As a result, federated learning is a better choice than any other method in finance and healthcare-related businesses, and on mobile devices and IoT systems.

8.2.1 Centralized Learning

In traditional centralized methodology, all training data is assembled and stored on one server or a homogeneous data center cluster. All the collected data is then loaded to a single model that gets trained with the help of batch or stochastic descent. This model can work across the whole dataset, which means that it can be developed highly optimally and without errors. However, such an approach requires offloading considerable amounts of data from the sources, involves concerns about data privacy due to aggregation of data, and creates vulnerabilities. Additionally, if the data is very big, or regulations prevent data transmission, it might not be feasible for large scale real application.

8.2.2 Distributed Learning

Distributed learning, often confused with data parallelism training, is defined as a technique that involves the division of raw enormous data between several information units, such that while the large data is held centrally, it is divided into several parts and parallelly trained on the replicas of model in different data pieces. Gradient updates are made synchronously (all-

reduce) or asynchronously between the nodes. Although it is arguably the fastest way to work with large data, its premise makes it depend on the proper integration and availability of data. Distributed learning is mostly regarded as an engineering optimization approach rather than a data security measure.

8.2.3 Federated Learning

Essentially, Federated education is about viewing the relationship between the data and the model differently. Information is treated as an independent being that cannot be removed from the device on which it was produced. The model does not consume the data; on the contrary, it travels to the data. This turning upside down is important in terms of privacy, compliance, and artificial intelligence system design. The ideas are discussed and contrasted in detail in table 8.1 for the listed attributes.

Table 1: Comparison of Learning Paradigms

Attribute	**Centralized Learning**	**Federated Learning**	**Distributed Learning**
Data Location	Central server	Client devices only	Distributed nodes
Privacy	Low (raw data shared)	High (no raw data shared)	Medium
Communication	Low overhead	High (model updates)	Moderate
Scalability	Moderate	High	High
Control	Single authority	Coordinated	Autonomous
Fault Tolerance	Low (single point)	High (partial failures OK)	High
Use Case	Controlled environments	Healthcare, IoT, Finance	Research clusters

8.3 FEDERATED LEARNING: FOUNDATIONAL PRINCIPLES

A distributed optimization problem can be defined as federated learning. Suppose we are given a number of K clients that obtain a normal dataset Dk consisting of nk samples. In this case, our task consists in minimizing a global loss function F(w) such as:

$$\mathbf{F(W)}=\sum_{k=1}^{k} \frac{n_k}{n} \cdot F_{k(w)} \quad \text{..... (i)}$$

where n is equal to the sum total number of instances given as nk, and F(yk,w) is given as (nk) y=1Σl(xi,yi;w), where the challenge is how to minimize the global objective function without ever having to create the global data.

8.3.1 The Federated Averaging Algorithm

The process of implementing objectives using the Federated Averaging (FedAvg) algorithm, which was introduced by McMahan, is as follows: The server picks a group for every round of communication t, St, consisting of clients and sends out the present weights of the global model

wt. Each chosen client k processes E epochs of stochastic gradient descent over its data locally to produce new weights wtk. The server then brings together these changes:

$$\mathbf{W_{(t+1)}} = \sum_{k \in S_t}^{k} n_{\frac{k}{n}} \cdot w^{tK} \ldots\ldots\ldots \text{(ii)}$$

With this weighted average method, we obtain both efficient calculations and easy communication, since only O(|w|) data is needed for each client in each round. The convergence of the algorithm has been demonstrated under mild assumptions on the diversity of the data and on the local computation.

8.3.2 Cross-Device vs. Cross-Silo Federation

Federated learning (FL) can apply to two main contexts: the cross-device context and the cross-silo context. Different scenarios have variations in clients involved, their computing capabilities, communication infrastructure, and deployment requirements. It is important to understand the differences present in order to use suitable algorithms and communication methods in different contexts. Cross-device federation involves a very large number of various types of devices including smartphones, tablets, smart wearables and IoT sensors. In general, devices may have limited computing abilities, small memory, battery constraints and opportunities for connectivity. Disconnections may occur which results in a continued churn of clients and makes client involvement changing. This results in a necessity of using effective selection methods, methods of communication and aggregation that are able to manage unreliable computing devices. In comparison, the number of organizations which are part of the use of cross-silo federated learning is small but stable e.g., hospitals, banks, universities, research labs and governmental institutions. Most of them have reliable connections, powerful computing infrastructures and efficient data storage facilities. Unlike the cross-device cases, where sometimes cross-device learning is used, the organizations taking part in the process can be available during the training process, which permits more frequent synchronization and participatory model improvements. Cross-silo FL has gained popularity in those sectors where organizations want to train ML models together while complying with the existing regulations. The decision between cross-device and cross-silo federation relies on the domain of application and objectives of the system. Cross-device FL emphasizes scaling, communication efficiency, and reliability in dealing with untrustworthy clients whereas cross-silo FL emphasizes the importance of secure communication and model accuracy as well as data privacy. Both models play a fundamental role in the implementation of federated learning in different applications in practice.

8.4 SYSTEM ARCHITECTURES AND NETWORK TOPOLOGIES

The design of a federated learning architecture plays a significant role in how well it can scale and perform in a fault-tolerant and secure manner with effective communication between nodes. Therefore, there are three key topologies that have been developed and applied: centralized star, multi-tier hierarchical, and decentralized peer-to-peer.

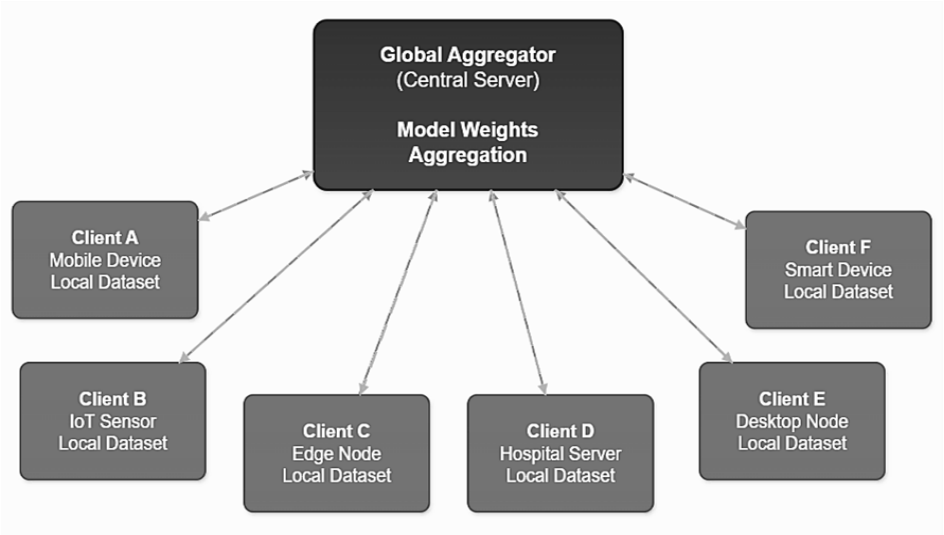


Figure 2 Federated Learning System Architecture — Client devices maintain local datasets while transmitting only model updates to the central aggregator.

8.4.1 Centralized Star Architecture

The concept of federated learning (FL) is one of a centralised paradigm in which a central node called a parameter server or aggregation server oversees the training process. When the global model has been initiated by the server, it is dispatched to a given number of clients for local training. Clients make use of the local data to update the model, thus sending back only the model parameters or gradients to the server while the local data is retained by clients. The server duly receives the updates in form of algorithms, for instance, Federated Average (FedAvg) and constructs a new global model to be sent for the next rounds of training. The centralized structure has many advantages like simple coordination, implementation, and communication with the aggregating models. The clients use one server which makes it easy to keep up with versioning and coordination. However, the problems arise with millions of clients connecting to the central server, which might create latency and overload issues. In addition, it can be seen as a single point of failure because if the server collapses or is hacked, the whole training process will suffer. Nevertheless, large companies like Apple and Google have been able to implement a centralized architecture in their applications like Gboard, Siri, and personalization of the keyboard thanks to their powerful server infrastructure and communication protocols.

8.4.2 Hierarchical Multi-Tier Architecture

The traditional centralized federated learning model has certain limitations when it comes to scalability and communications. As a solution to this, hierarchical federated learning (HFL) has been proposed. HFL is similar to centralized federated learning in that it employs a central server, but also incorporates one or more levels of intermediates so that model updates can be aggregated on edge servers, regional centers, cloudlets, etc. HFL allows local training by clients with model updates sent to the local edge aggregator. The edge server then aggregates the updates from different clients to get the regional model. The use of this multi-tier aggregation technique can significantly reduce communication delays, network load and bandwidth usage

due to the reduction of data transmitted across distant links. This is particularly effective for widely distributed networks with many devices connected in various network domains. Hierarchical federated learning is well-suited for smart cities, IoT systems, mobile edge computing, and large industrial networks, as edge servers are able to manage the local devices before sending data to the central one. Although it yields benefits, the HFL entails an additional level of architectural complexity. The most important characteristics of multiple aggregation levels are effective synchronization of resources, swift communication between the client, edge server and global server, and resource management. Furthermore, trust towards intermediates becomes a challenge because intermediates need to conduct secure model modification, among other things. Nonetheless, hierarchical FL is an optimistic way to solve the problem of FL scalability, reduce problems related to communication, and enable FL implementation in large and diverse settings.

8.4.3 Decentralized Peer-to-Peer Architecture

Unlike conventional federated learning (FL), decentralized federated learning (DFL) does not require a centralized server for data aggregation. In fact, every client trains on its data and directly shares the updates of its model with other clients without relying on a reliable intermediary. Instead of working with a trusted central authority, DFL clients use peer-to-peer communication, gossip protocols or structured overlay networks to transfer the information necessary. More specifically, the local model training is conducted by the clients independently based on their personal data and periodically relative to the communication graph. Therefore, peers may exchange the updates of their models based on the required model exchange graph and use the updates for local aggregation without the need for a central system. One of the major advantages of decentralized Federated Learning (FL) is its excellent fault tolerance and reliability which helps overcome the issues of failing entities. In addition to that, the learning process can continue even if some participants are unreachable or there are communication failures. The underlying architecture safeguards dependency on centralized facilities thus it can fit well in the circumstances of distributed platforms, i.e., Internet-of-Things (IoT) networks, vehicle fleets, wireless sensor networks, and edge computing setups. In addition to that, decentralization of FL helps enhance privacy by obviating the need for trusted aggregation servers. The decentralized structures, however, bring a few difficulties. Consistent synchronization of the model becomes more difficult when many clients are involved compared to centralized systems, which can result in communication overhead if the same data is shared with clients repeatedly. The necessity to keep convergence in a dynamic network, work with heterogeneous data distribution and coordinate the models requires advanced optimization methods. New approaches like Gossip FL and Epidemic FL have proven to achieve convergence comparable with centralized FL even in cases of non-independent and identically distributed (non-IID) datasets. These advances show how promising decentralized FL is for solving the problems of scalability and reliability in collaborative machine learning without breaching confidentiality.

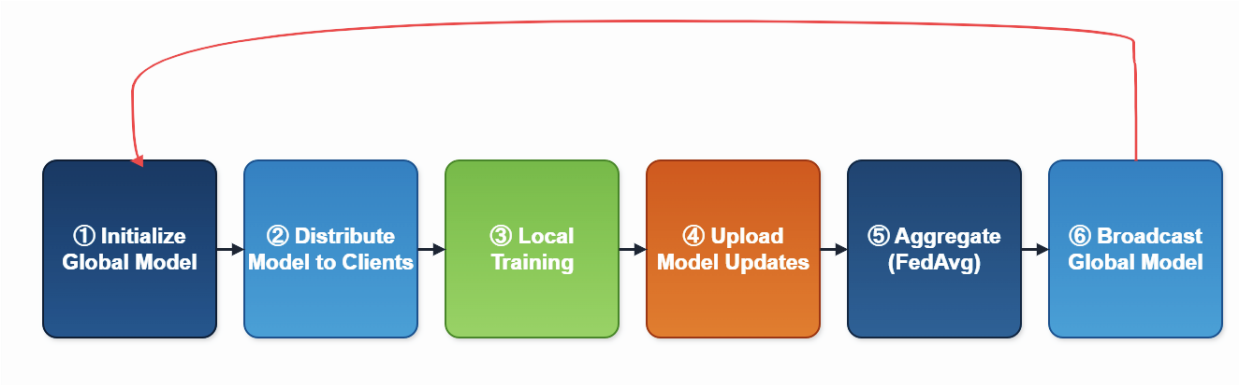


Figure 3: Federated Learning Workflow — The iterative cycle of model distribution, local training, update aggregation, and global broadcast repeated across communication rounds.

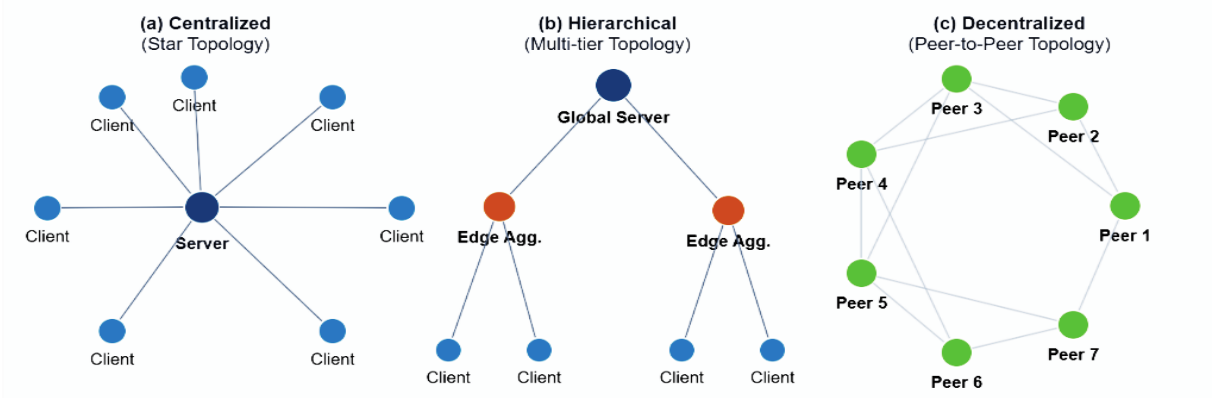


Figure 4. Federated Learning Network Topologies — (a) Centralized star, (b) Hierarchical multi-tier, and (c) Decentralized peer-to-peer configurations.

8.5 Aggregation Algorithms

The aggregation algorithm acts as the mathematical foundation of federated learning, guiding how model updates computed locally comprise a meaningful global model. The selection of an aggregation method heavily influences the speed of convergence, the robustness to heterogeneity, and the effectiveness against malicious participants.

8.5.1 Federated Averaging (FedAvg)

As explained in Section 8.3.1, FedAvg averages client model weights in a weighted manner that is proportional to the size of the corresponding local dataset. While thus effective in practice under IID conditions, FedAvg also has been demonstrated to experience the issue of objective inconsistency, which is the difference between local and global objectives when distributions of data are significantly different across clients. In this case, this issue is termed "client drift," which leads to difficulties in convergence and lowering the quality of the global model.

8.5.2 FedProx

FedProx solves the problem of client drift by introducing a proximal term to the local optimization objective function. Every client works on minimizing:

$$w_{t+1}^{(k)} = \arg\min_{w} \left\{ F_k(w) + \frac{\mu}{2} \|w - w_t\|^2 \right\} \quad \text{.....(iii)}$$

where μ is the hyperparameter which regulates the proximity to the global model. The proximal regularization assures the local models do not stray too far from the global optimum, making convergence more stable on heterogeneous datasets. It has been shown that FedProx performs better than FedAvg in the cross-silo scenario characterized by high data heterogeneity.

8.5.3 SCAFFOLD

SCAFFOLD (Karimireddy) makes use of control variates to handle the drift attributable to the client but does not involve the effort of proximity term as necessary. It guarantees convergence without depending on data heterogeneity and also works with communication complexity as in FedAvg as it keeps variance reduction variables on both server side and client side. The only major limitation is that it requires additional memory for storage of the control variates, something which may not be feasible for devices having constrained memory.

8.5.4 FedNova and Beyond

FedNova resolves the problem of objective inconsistency through gradient normalization among clients who have undergone varying numbers of local updates. Also, more recently, some algorithms such as FedDyn, MOON, and FedBE present methods of dynamic regularization, model contrastive learning, and Bayesian ensemble aggregation respectively. The choice of a particular algorithm will depend on its characteristics regarding deployment features like the extent of data heterogeneity or client computational power and communication budget.

8.6 PRIVACY AND SECURITY MECHANISMS

Due to the local storage of raw data, FL has an inherent advantage in terms of privacy which is, however, not fully guaranteed. Numerous attacks, such as gradient inversion, model inversion, and membership inference, have demonstrated that updates of the model can reveal critical information about its underlying training data. Thus, it is essential for an efficient FL environment to employ additional methods aimed to prevent privacy issues and safeguard against adversarial agents.

8.6.1 Differential Privacy

An approach called Differential Privacy is one that is built on mathematics and offers secure privacy. The algorithms that are said to be following this approach have (ε, δ)-DP and work on different neighboring databases 'D' and 'D' such that they differ at least with one data entry:

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in S] + \delta \quad \text{.....(iv)}$$

In the field of Federated Learning, Differential Privacy is achieved by following some steps like clipping stats of the model's gradient and adding noise in the form of Gaussian or Laplace distribution to the updates before sending them to servers. The ε privacy term is used to control

the level of privacy offered by the algorithm which means that when ε is kept low it offers stronger privacy but at the same time decreases the performance of the model.

8.6.2 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) allows different parties to compute a function together, such as model aggregation, without having to reveal their inputs to each other. One of the SMPC protocols employed in FL, namely additive secret sharing, makes it possible to compute the sum of model updates without the aggregator learning individual clients' updates. Although SMPC offers cryptographic privacy guarantees similar to those of DP, the former has higher communication and computation costs and should be mainly used in inter-silo environments where the number of participants is limited.

8.6.3 Homomorphic Encryption

Homomorphic Encryption (HE) refers to carrying out computation directly on the ciphertext. The result would be plain text as we expected. FL clients have the option to encrypt their model updates before sending them to the server, which in turn can sum these ciphertexts without decrypting them. Fully Homomorphic Encryption (FHE) schemes are still too costly in terms of computation resources, while partially homomorphic schemes like Paillier's cryptosystem enable sufficient computation for additive operations, which are needed in the process of gradient aggregation.

8.6.4 Byzantine Fault Tolerance

Malicious clients may attack open federated systems in order to undermine their impact on the global model. Average approaches have been replaced with Byzantine-robust algorithmic schemes such as Krum, MultiKrum, coordinate-wise median, and trimmed mean. Krum picks only one client update, while coordinate-wise median gets the median for each coordinate instead of taking a median across values. It can withstand up to (K/2 - 1) Byzantine clients under special conditions. The challenges faced with the robust aggregator are in the form of computational cost incurred, and potential loss of accuracy on benign data.

8.7 KEY CHALLENGES IN FEDERATED LEARNING

8.7.1 Data Heterogeneity and Non-IID Distributions

One of the major problems in real-life federated learning implementations is the lack of IID (independently and identically distributed) data among clients. A hospital serving only elderly patients will experience an entirely different clientele from a hospital that serves only kids. This is also true for the language users type in an application such as a mobile keyboard; this is not done in identical setups; language behaviors and patterns are actually influenced by various factors such as the age of users, their geographical location, as well as their work-related background. The non-IID character of federated data may lead to inaccurate gradient estimations and may result in the impossibility to converge to the optimal solution.

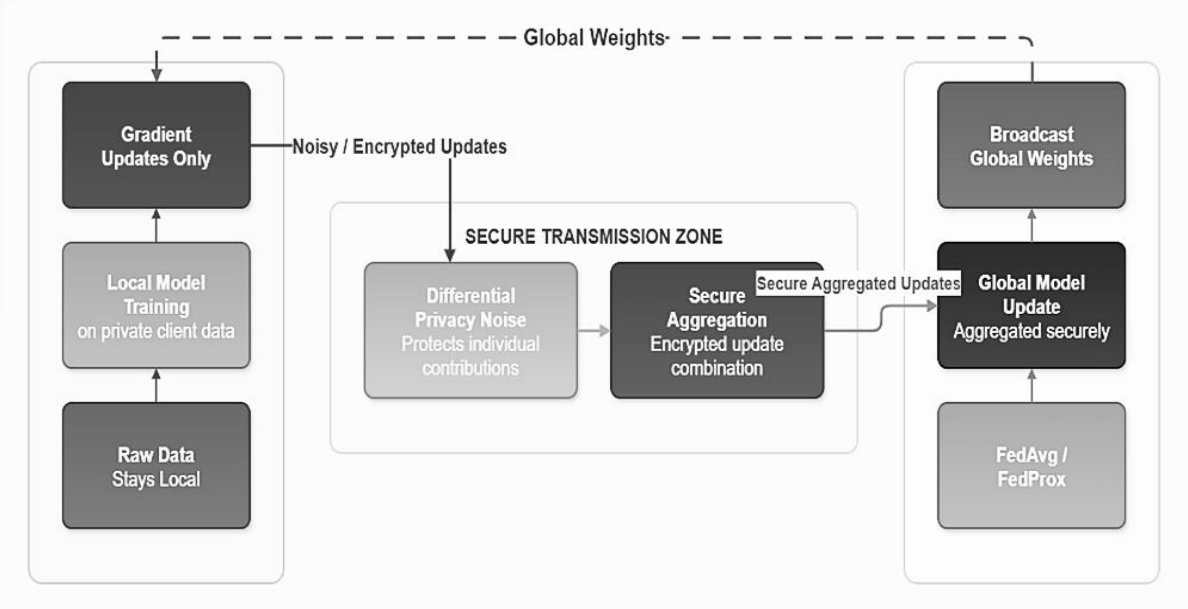


Figure 5. Privacy and Security Model in Federated Learning — Raw data remains sovereign at the client; only differentially private gradient updates traverse the secure transmission zone.

The role of experimental design in measuring and handling non-IID-ness cannot be overemphasized. Typically, the heterogeneity is modeled by using Dirichlet distribution where the concentration parameter α indicates the magnitude of skewness. Figure 8.6 illustrates typical non-IID distributions for six federated clients, which are characterized by substantial data imbalance complicating the task of aggregation.

8.7.2 Communication Efficiency

The communication cost is one of the major bottlenecks in federated systems. A contemporary deep neural network can be made of up to two hundred million parameters and each update set per customer in every round can mean that the amount of transmitted data can be measured in megabytes. If we consider many clients and numerous rounds, the total volume of data transfer can be quite significant, especially for bandwidth-limited devices, like mobile phones. There are three main strategies to tackle this issue: Gradient compression involves techniques such as Top-K sparsification, random sparsification, and structured dropout which send only the most relevant components of the gradient and thus reduce communication per round by 10 to 100 times while incurring only a small degradation in accuracy. Quantization decreases the communication bandwidth up to 4-32 times which reduces the sizes of the gradients from 32-bit floationg point numbers to 8-bit integers or binary formats. Quantization aware training can be used to neutralize the noise produced by quantization. Local Computation Amplification: It is implied that more local SGD steps E done before communication impacts the speed of model synchronization, yet may lead to an increase of client drift.

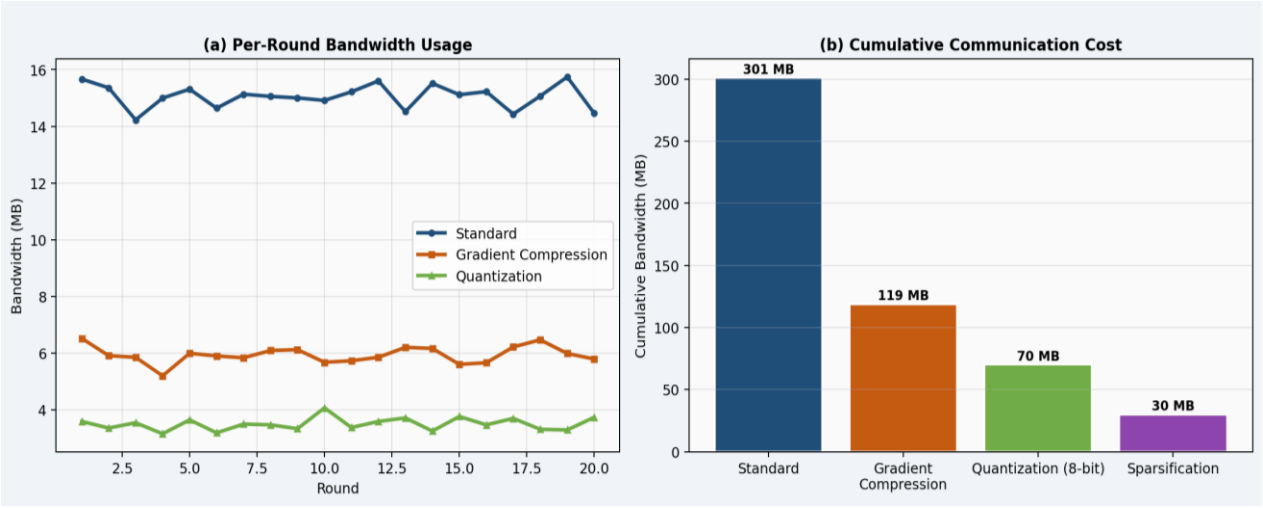


Figure 6. Communication Cost Across Training Rounds — Gradient compression and quantization strategies reduce cumulative bandwidth by up to 75% compared to standard updates.

8.7.3 System Heterogeneity and Stragglers

The performance of federated client pools, based on the hardware capabilities, connectivity characteristics and battery levels, varies a lot. In synchronous FL protocols, the global round is synchronized based on the slowest responding client which may result in long delays in the case the slowest client can be characterized as a straggler. The use of asynchronous FL protocols helps increase the efficiency because the server is able to accumulate the updates as they arrive, but there is a risk of receiving outdated updates from clients falling behind. Semi-synchronous aggregation combined with the strategies based on the selection of fast clients provides a practical solution. Moreover, the efficiency of the system is increased thanks to the adaptive client selection approaches that utilize both the expected training time and the quality of training data used.

8.7.4 Model Convergence Under Partial Client Participation

In a scenario involving widespread cross-device implementation, reaching out simultaneously to all the clients every time becomes challenging. In such conditions, only a limited number of clients C can be chosen with $C < 1$ for each round. It is common to make random selection for this purpose, which might add some variance in gradient estimations. Accordingly, if the selected set of clients is not truly representative of the entire client pool, it may increase the uncertainty of the model — a situation wherein we are able to minimize the uncertainty by picking up more informative data again tends to speed up convergence, but at the same time, it may rise representation bias. Convergence rates for partial participation have been theoretically derived with different data heterogeneity conditions.

8.8 REAL-WORLD APPLICATIONS OF FEDERATED LEARNING

8.8.1 Healthcare and Biomedical Research

Federated Learning has the potential to be used in a variety of applications and sectors, such as healthcare and biomedical research. The healthcare sector generates massive amounts of sensitive data such as electronic health records (EHRs), medical images, laboratory results, and genomic data. This kind of data usually remains under strict limits of collaboration among clinics and research centers for privacy and ethical reasons. Therefore, federated learning is applied to enable other healthcare institutions to build machine learning models without transferring their clients' data.

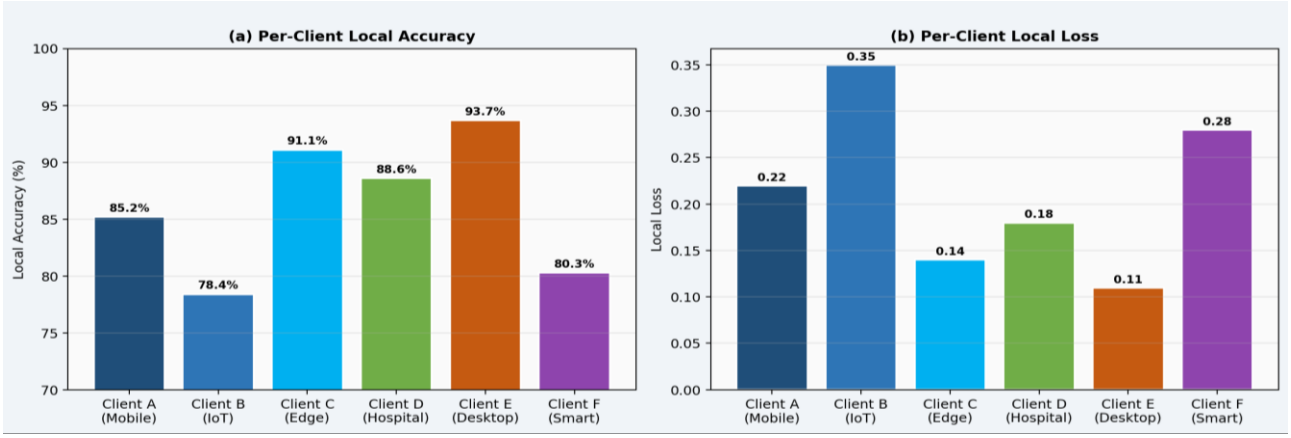


Figure 7. Per-Client Performance Comparison — Local accuracy and loss vary significantly across client types, reflecting system and statistical heterogeneity.

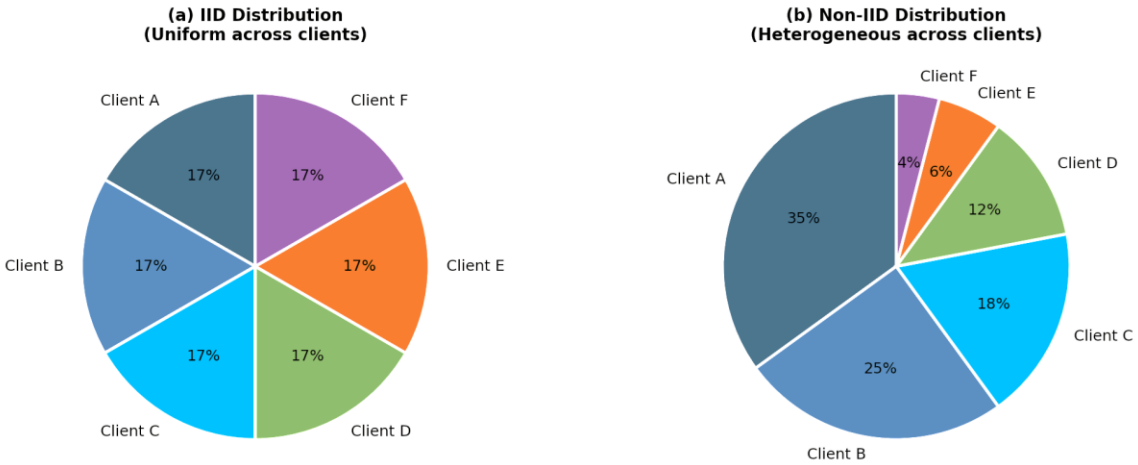


Figure 8. Data Distribution Across Federated Clients — Contrasting IID (uniform) and Non-IID (heterogeneous) data partitioning strategies.

One of the significant benefits of FL is its ability to improve the model's performance by making use of various datasets obtained from different organizations. However, the generalizability of locally trained models may be limited due to the characteristics of patient populations, the prevalence of diseases, and the availability of imaging devices at different facilities, resulting in some medical datasets gained from such facilities that are limited in reach. Moreover, FL makes it possible to create stronger and more reliable diagnostic software due to the model updates being combined rather than the actual data being aggregated and the privacy of the patient being protected. It has been shown that federated learning is uniquely effective when applied

to various medical tasks, such as disease diagnosis, medical imaging analysis, cancer diagnosis, radiology interpretation, pathology, and personalized medicine. The MELLODDY consortium is a case study that features collaboration among ten drug companies who utilize federated learning to advance the process of drug discovery while preserving proprietary data. Thanks to this partnership, significant improvements in predictive performance was achieved, without compromising proprietary information. Nonetheless, healthcare FL has its concerns as well, such as data diversity, communication overload, model safety, and compliance issues. Nevertheless, the continued advancement of privacy-preserving techniques and distributed optimization has made federated learning an increasingly desirable technology in the context of safety and collaborative healthcare research.

8.8.2 Mobile and Edge Intelligence

Mobile and edge intelligence is the most widely used instance of federated learning in the real world. With modern smartphones, tablets, and other edge devices, huge amounts of user-specific data are produced constantly through typing, speaking, surfing the internet, and using apps. This data is often private and sending it to central servers raises serious privacy issues. Federated learning allows devices to train ML models under conditions that keep the data on their own devices and thus they are more privacy-friendly and less susceptible to data leakage. Google introduced a revolutionary application of federated learning (FL) to develop its next generation of word prediction systems in Gboard, its virtual keyboard application. When users type on their Android devices, millions of devices collaborate to build language models. Only encrypted model updates are sent to the central server, which receives all updates. Thus, the user does not need to fear having his data revealed and is ensured improved reliability and personalization in typing. Also, Apple applies federated learning to improve Siri's speech recognition abilities and the suggestions provided by QuickType as well as other smart applications available on its devices. The systems function because of the ability to learn from multiple user interactions, without violating privacy and reducing costs of data transfer. The broader use of federated learning within mobile environments has led to advancements in various aspects such as client selection methods, secure aggregation, differential privacy, communication-efficient algorithms, and power-saving model training. Owing to the emergence of mobile computing and edge intelligence, federated learning will become an important technology to enable privacy-preserving, intelligent and personalized services for billions of IoT devices around the world.

8.8.3 Financial Services and Fraud Detection

Federated learning has another significant application, which is in the financial services ecosystem where security, privacy, and compliance are critical concerns. Financial institutions like banks, insurers, payment systems, and FinTech companies are known to store and process large amounts of confidential data. This data includes transactions, account history, credit history, and investment details. Such data is generally not shared among institutions, which can create challenges in developing a joint model. However, federated learning helps with this issue by enabling organizations to collaboratively create machine learning models while preserving confidentiality of the data involved. Fraud detection is one of the key uses of federated learning (FL) in finance. New financial fraud schemes keep appearing, and models developed from the data of one institution may not be able to detect a new fraud. FL helps improve the effectiveness

of fraud detection systems, while at the same time protecting the private data of customers, as it enables the sharing of information about transactions among various institutions. FL is also used for credit risk assessment, for anti-money laundering (AML), for customer segmentation, and for financial prediction, letting financial organizations have more available information without breaking the data privacy regulations. An intriguing use case outside of academia is the successful implementation of federated learning in financial applications that include credit risk assessment in China by WeBank. The Federated AI Technology Enabler (FATE) framework has been developed as a method that allows secure collaboration in the financial services domain. With the use of FATE, organizations can implement secure model aggregation, encrypted computation, and distributed learning. Thus far, federated learning has been very useful in a well-regulated environment. In the future, federated learning may play an increasingly important role in the development of secure and privacy dating solutions.

8.8.4 Autonomous Vehicles and Connected Transportation

The use of Federated learning (FL) appears to benefit autonomous vehicles and intelligent transport, which are operated by sophisticated machine learning techniques for the purposes such as perception, object detection, lane identification, sign recognition, and decision making. These models should be trained on multiple scenarios to be working well with different weather, conditions of the road, traffic patterns, and other emergencies. However, this is not possible for one manufacturer or fleet operator to collect enough data to cover driving situations. In this respect, federated learning is a good option since it enables different vehicles and companies to collaborate in improving joint models while keeping their data private on board of their vehicle. AVs consistently generate enormous data volumes using their cameras, LiDAR, radar, GPS, and other vehicle-integrated sensors. Data collected by these sensors can be sent to central servers, but this is usually associated with problems such as bandwidth consumption, privacy issues, security breaches, and proprietary information management. In the federated learning method, several vehicles are in charge of training their models with their own driving data, and the only information transferred to the main server is information about the model. This joint approach ensures that multiple models are trained and updated without any major expenses and security problems. Many of the positive outcomes of federated learning are recorded in the development of perception systems, traffic foreseeing, collision avoidance, and cooperative driving intelligence. In the autonomous driving area, Intel's OpenFL and Openminded PySyft are engaged in research of the federated means of training perception models. Experimental results demonstrated that models trained via federated approach cope with recognizing infrequent driving events and dangerous situations better than the models trained on the isolated data sets. In the future, federated learning will play a major role in the development of safe and trustworthy autonomous driving systems and the good cooperation between car-makers, operators, and research institutes because connected cars and smart infrastructure are developing increasingly.

8.9 EMERGING RESEARCH DIRECTIONS AND FUTURE CHALLENGES

8.9.1 Personalized Federated Learning

Personalized Federated Learning (PFL) is a unique type of federated learning that was created in order to improve the performance of the model for its respective client while client data is not identical. In traditional federated learning, the global model is created by merging updates received from different clients. However, the global model cannot be applied to every case since the data of each client is often completely different or not independent and identical (non-IID). PFL solves the problem of developing personalized models that serve the individual needs and characteristics of each client while training with the information received from other clients. Several personalization techniques have been proposed which aim to balance worldwide knowledge acquisition with local customization. Usually, this is done through local fine-tuning in which users obtain access to the common model and continue its training with their unique local dataset. Another solution might be "layerwise personalization" of the network: in lower layers the architecture learns features that are commonly shared between users while layers higher are dedicated to features that are particular to the customer. In the federated learning environment different meta-learning techniques have been adopted, based on the concept of Model Aware Meta-Learning, for example Per-FedAvg, which helps adapt models to customers' unique dataset in no time with minimum local updates. These techniques have been demonstrating much higher performance in terms of both accuracy and robustness when compared with traditional federated learning methods. Even if personalized federated learning brings numerous advantages, it does come with its unique challenges that include selecting the appropriate personalization approaches, defining both the local and global objectives, and the risk of overfitting data from small local data sources. Among the challenges currently being studied are the efficiency of communication, safeguarding privacy, as well as ensuring fairness for clients using the model. Given the growing demand for customized intelligence in sectors such as healthcare, recommendation systems, mobile applications, and smart services, the role of pFL is about to rise significantly.

8.9.2 Split Learning

Split learning technique may be referred to as a joint learning approach to lessen the strain on computing power of client devices while safeguarding the privacy of data. It is important to note that split learning is not a synonym for federated learning since, in the latter technology, each client has an opportunity to address his or her complete machine learning model individually. Split learning differs from conventional federated learning process in the sense that the model is being split at the level of its first few layers which are trained with clients’ data while the rest of the model is taking place on the side of the server. As a result of this approach, even devices which have limited capabilities can participate in the joint learning process. During training, the client transmits the forward pass across its fraction of the network and sends the intermediate representation of the model, referred to as smashed data, to the server. The server computes the loss after the forward pass and then performs the backpropagation task before transmitting the gradients back to the client to update the local layers. Split learning can provide some privacy protection since it sends only intermediate results instead of raw training data and reduces communication of sensitive information. Split learning has advantages but also has some disadvantages. It is possible to use only small parts of the identifying activation functions; however, this is enough to get some information about the original dataset, which is a situation that raises certain concerns regarding privacy. Moreover, training delays and communication overhead are expected to increase between the client and server. In order to solve these

problems, researchers propose hybrid architectures based on split and centralized learning methods that can enhance computational effectiveness, scalability, and reduce privacy issues for different applications.

8.9.3 Federated Learning with Foundation Models

The development of foundation models including large language models (LLMs) and vision transformers (ViTs) has opened new opportunities and challenges for federated learning (FL) in recent years. These models are very potent and suited for many different natural language processing and computer vision applications, but they can have millions, if not billions, of parameters. Such computations may be highly demanding or at times even impossible on resource-constrained devices such as mobile phones, IoT devices, and edge-computing platforms. Thus, the researchers are exploring ways of achieving efficient collaborative modeling without incurring high computational or communication costs. A potentially compelling alternative is parameter-efficient fine-tuning (PEFT), which consists of fine-tuning only a chosen subset of model parameters, as opposed to the whole foundation model. Low-Rank Adaptation (LoRA), Prompt Tuning and Adapter Layers are some techniques that enable memory, communication and power efficiency while ensuring competitive performance. These techniques let edge devices participate in federated training without requiring high-end equipment or large bandwidths. Federated learning potentials using foundation models have multitasking capabilities including creating personalized virtual digital assistants, AI in medicine, recommendation systems, translations across languages, and conversational artificial intelligence platforms that can function confidentially. The positive side of federated instruction tuning lies in earning specialization of large language model to serve an individual and assigned task without the need of data transfer to the clouds. However, despite the great prospects of the area, it suffers from challenges, including communication inefficiency, non-homogeneous models, security, fairness, and aggregation of the parameter updates.

8.9.4 Incentive Mechanisms and Fairness

The success of federated learning highly relies on the active participation of clients who share their quality data and computational resources during training. However, the participants in the training process incur expenses related to computation, energy utilization, storage, and communication, so effective incentive mechanisms must be developed in order to push them towards continued participation. Thus, the design of both fair and privacy-ensured compensation systems becomes a highly relevant area of research, particularly in light of the reference to the coordination of efforts of multiple independent organizations in various fields of activity, such as healthcare, finance, mobile networks, and Internet of Things (IoT), while keeping privacy of the data intact. Various ideas have been proposed concerning how the contributions of the clients can be evaluated and rewarded. Game-theoretic models focus on finding ways to stimulate cooperative behavior of participants by creating mechanisms that prevent their noncooperative behavior. The application of game-theoretic methods that use the Shapley value known from cooperative game theory helps to compute the contribution of certain players by identifying how their presence in the model will influence its performance. Background analysis of reliable participants as well as ensuring trustful cooperation among participants during the cooperation process has resulted in the implementation of reputation-based systems. In addition to incentives, another important factor in federated learning is

fairness. Customers often have datasets that vary in size, quality, and demographics, which causes the model to perform differently for each participant. For businesses with less or less representative data, an optimization that works for the majority of clients might not work as well. In order to guarantee that various customers meet the optimization objective accurately and fairly, fairness-aware optimization algorithms are being developed. Enhancing data value, federated optimization's fairness, and participant incentive mechanisms would increase participation, foster trust, and make it possible for federated learning to be adopted in large-scale collaborative AI systems in a sustainable manner.

CHAPTER 9

BLOCKCHAIN-ENABLED FEDERATED INTELLIGENCE

Contributed by: Bishal Ray, Sumana Chakraborty, Tulika Paul

9.1. INTRODUCTION

9.1.1 The AI Paradox: Data Utility vs. Privacy

In the contemporary digital era, Artificial Intelligence (AI) and Machine Learning (ML) have fundamentally transformed how complex systems operate, ranging from predictive healthcare diagnostics to autonomous transportation. Traditionally, training robust AI models required the aggregation of massive volumes of raw data into centralized repositories or cloud servers. However, this centralized paradigm has precipitated a critical paradox: the inherent conflict between data utility and data privacy. As edge devices—such as smartphones, IoT sensors, and smart wearables—generate unprecedented amounts of highly sensitive personal data, centralizing this information poses severe ethical, legal, and security risks. The introduction of stringent data protection frameworks, such as the General Data Protection Regulation (GDPR) and various regional data privacy laws, has made traditional data-harvesting practices legally complex and financially risky.

9.1.2 The Evolution: From Cloud AI to Federated Learning

To address this data privacy bottleneck, Federated Learning (FL) emerged as a revolutionary paradigm shift. Introduced primarily to decentralize the training process, FL operates on a simple yet profound principle: bring the model to the data, rather than bringing the data to the model. In a standard Federated Learning ecosystem, a central server distributes an initial, baseline ML model to various participating nodes or edge devices. These devices train the model locally using their own private datasets. Instead of transmitting raw data back to the server, the devices only compute the mathematical updates (gradients or weights) and send these parameters back. The central server then aggregates these updates—often using algorithms like Federated Averaging (FedAvg)—to create an improved global model. This approach significantly mitigates privacy risks, as the raw data never leaves the local device.

9.1.3 The Trust Gap: Vulnerabilities in Traditional FL Architectures

Despite its privacy-preserving architecture, traditional Federated Learning is not without critical flaws. A rigorous analytical view reveals that standard FL operates on a "star topology," which heavily relies on a single central aggregator. This reliance creates a significant

vulnerability known as the single point of failure (SPOF). Firstly, the central server must be considered "honest-but-curious." Participants have no choice but to blindly trust that the aggregator will not attempt to reverse-engineer their local updates to infer sensitive information, a vulnerability often exploited through Model Inversion or Inference attacks. Secondly, the centralized nature makes the system highly susceptible to adversarial attacks. If a malicious actor compromises the central server, or if rogue edge devices inject intentionally corrupted model weights—known as Data Poisoning or Model Poisoning the integrity of the entire global model is compromised. Therefore, while FL solves the data privacy issue, it inadvertently creates a massive trust gap. The system demands decentralized execution but paradoxically relies on centralized trust.

9.1.4 Decentralizing Trust: The Convergence of Blockchain and Federated Intelligence

To bridge this trust gap, researchers are increasingly turning to Distributed Ledger Technology (DLT), specifically Blockchain. Blockchain offers a mathematically secure, immutable, and decentralized infrastructure that can fundamentally replace the vulnerable central aggregator in an FL network. When Federated Intelligence is integrated with Blockchain, the central server is replaced by a decentralized network of validating nodes. Local model updates are no longer sent to a single entity; instead, they are broadcasted to a peer-to-peer (P2P) network. Smart contracts—self-executing code embedded within the blockchain—automatically verify the quality of the incoming updates, execute the aggregation process, and record the immutable global model on the ledger. This convergence creates a "trustless" environment. Participants do not need to trust a central authority; they only need to trust the cryptographic consensus mechanism.

9.1.5 Chapter Scope and Objectives

Recognizing the transformative potential of merging Distributed Ledger Technology with decentralized machine learning, the primary scope of this chapter is to provide a comprehensive and critical analysis of Blockchain-Enabled Federated Intelligence. Moving beyond purely descriptive concepts, this manuscript aims to deconstruct the architectural intricacies, security enhancements, and operational trade-offs inherent in this hybrid paradigm. To achieve a rigorous and analytical exploration of the subject, the core objectives of this chapter are structured as follows:

Critical Evaluation of Existing Paradigms: To critically evaluate the vulnerabilities of traditional Federated Learning, specifically focusing on the single point of failure, data poisoning attacks, and the inherent trust deficit present in centralized aggregators.

Architectural Synthesis: To propose and detail a theoretical, blockchain-driven framework that leverages smart contracts for the secure, automated, and trustless aggregation of local machine learning models.

Security and Performance Analysis: To conceptually analyze how blockchain integration effectively mitigates adversarial threats while realistically assessing the computational overhead, storage constraints, and network latency introduced by decentralized consensus mechanisms.

Practical Implications and Future Trajectories: To bridge the gap between theoretical models and real-world application by exploring contemporary use cases (such as decentralized healthcare networks or smart IoT environments), thereby highlighting emerging research directions.

9.2. CONCEPTUAL FOUNDATION AND CRITICAL LITERATURE REVIEW

9.2.1 Federated Learning Taxonomy: Architectural Overview and Current Limitations

Federated Learning (FL) represents a paradigm shift in artificial intelligence, fundamentally decoupling the model training process from the need to centralize raw, sensitive data. The architectural taxonomy of FL is primarily categorized into three frameworks based on data distribution: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL). In HFL, multiple devices share the same feature space but possess different user samples, which is highly applicable in mobile keyboard predictions or regional healthcare diagnostics. VFL is utilized when collaborative institutions share the same user base but hold different data features, such as a bank and an e-commerce platform jointly training a risk model. FTL applies advanced transfer learning when neither the sample space nor the feature space overlaps significantly. Despite its privacy-preserving nature, the traditional FL architecture is fundamentally constrained by its reliance on a central parameter server. The standard workflow dictates that local devices compute gradients and transmit them to a central aggregator. This creates a severe Single Point of Failure (SPOF). If the central server experiences downtime, the entire global training halts. More critically, the central aggregator operates blindly, inherently trusting all incoming model weights. This trust loophole makes the network highly susceptible to Data Poisoning (where a node injects fake data) and Model Poisoning (where an adversary uploads corrupted gradients to derail the global model). Furthermore, recent studies demonstrate that curious central servers can reconstruct private local data from shared gradients using Model Inversion attacks.

9.2.2 Blockchain Fundamentals: Consensus Algorithms and Edge Device Relevance

To eliminate the central aggregator and its associated vulnerabilities, researchers have turned to Distributed Ledger Technology (DLT), specifically Blockchain. A blockchain provides an immutable, decentralized, and transparent ledger maintained by a peer-to-peer network. The core engine of any blockchain is its Consensus Algorithm, which ensures all nodes agree on the network's state without a central authority. However, integrating blockchain with Federated Learning requires careful consideration, as FL relies heavily on resource-constrained edge devices (such as smartphones and IoT sensors).

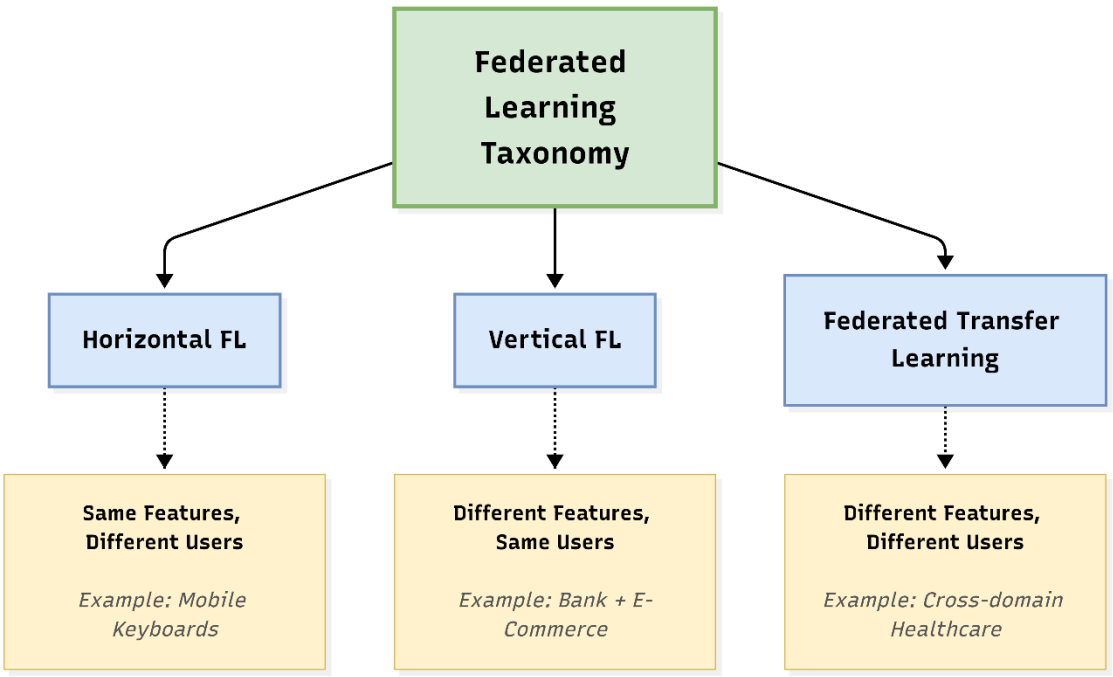


Figure 1: Architectural taxonomy of Federated Learning paradigms based on data partitioning strategies

Proof of Work (PoW): The foundational algorithm behind Bitcoin requires nodes to solve computationally intensive cryptographic puzzles. While highly secure, PoW is entirely impractical for edge-based FL, as mobile and IoT devices lack the massive computational power and energy reserves required for mining.

Proof of Stake (PoS): Instead of computational power, PoS selects validators based on the number of tokens they hold and are willing to "stake." This drastically reduces energy consumption, making it more edge-friendly. However, it can lead to wealth monopolization, where richer nodes dominate the model aggregation process.

Delegated Proof of Stake (DPoS): In this model, network participants vote for a small, trusted group of delegates who validate the blocks. DPoS significantly enhances transaction speeds and reduces latency, making it highly suitable for asynchronously aggregating large machine learning models.

Proof of Authority (PoA): PoA relies on identity and reputation rather than wealth or computing power. Pre-approved validator nodes secure the network. Because it demands minimal computational overhead and offers rapid block times, PoA is currently the most optimal consensus mechanism for permissioned, enterprise-level Federated Intelligence networks, such as inter-hospital medical imaging collaborations .

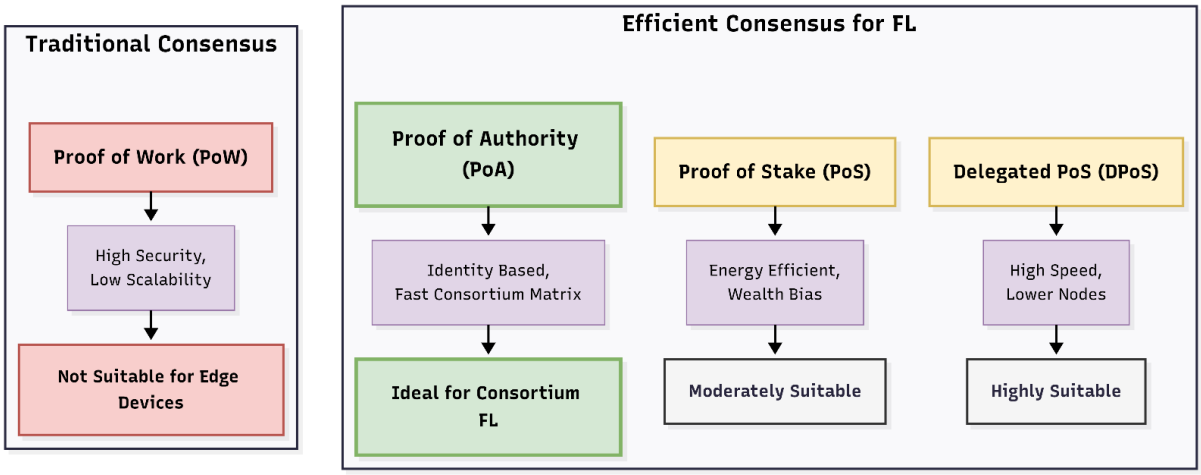


Figure 2: Evaluation of Blockchain consensus mechanisms and their applicability to edge-based Federated Intelligence network

9.2.3 Critical Synthesis: Methodological Trajectories in Recent Literature (2023–2026)

A rigorous review of contemporary literature spanning 2023 to 2026 reveals that the integration of Blockchain and Federated Learning is an actively evolving domain. Analyzing over a dozen recent high-impact studies, the existing research can be critically categorized into three dominant methodological trajectories.

- ***Incentive-Driven and Reputation-Based Frameworks:*** A major focus in recent literature is solving the "free-rider" problem, where nodes participate in the FL network to download the global model without contributing meaningful data. Recent frameworks introduce smart-contract-based incentive mechanisms. These contracts cross-validate local updates; if an update improves the global model's accuracy, the contributing node receives cryptocurrency tokens. Conversely, nodes submitting poisoned data face reputation slashing.
  - ***Critique:*** While economically robust, these frameworks often introduce prohibitive latency. Executing complex validation algorithms for hundreds of nodes directly on-chain creates a massive communication bottleneck, rendering it impractical for Deep Learning models with millions of parameters.
- ***Privacy-Enhanced Layered Architectures***: To counter gradient leakage vulnerabilities, recent studies have combined Differential Privacy (DP) and Homomorphic Encryption (HE) with blockchain networks. In these models, clients add mathematical noise to their weights before uploading them to the blockchain, or validators aggregate the models while they remain fully encrypted.
  - ***Critique:*** The primary limitation of this approach is the utility-privacy trade-off. Injecting heavy noise to secure the gradients often degrades the final accuracy of the AI model. Additionally, executing homomorphic encryption inside smart contracts drastically spikes the computational cost, often exceeding the capacity of standard edge devices.
- **Asynchronous and Scalability-Focused Protocols:** Recognizing the latency issues of block-based ledgers, a newer wave of research advocates for Asynchronous Federated Learning using Directed Acyclic Graph (DAG) structures instead of traditional blockchains. These

networks do not wait for all devices to finish training; they update the global model dynamically as new gradients arrive.

— ***Critique:*** While asynchronous DAG protocols resolve latency and easily accommodate devices with different processing speeds, they introduce a critical flaw known as "model staleness." Slow devices often submit updates based on outdated global models, which can severely disrupt the mathematical convergence of the final intelligence model.

Table 1: Comparative Analysis of Existing Frameworks

Methodological Paradigm	Representative Focus	Consensus Used	Security & Privacy	Scalability	Key Critical Limitation
Reputation-based FL	Addressing free-riders	DPoS	High	Moderate	High contract overhead
Differentially Private BFL	Deep Gradient Leakage	PoS	Very High	Low	Accuracy degradation
Homomorphic Encrypted BFL	Data confidentiality	PoA	Extremely High	Very Low	High computational cost
Asynchronous DAG-FL	Latency/Sync issues	DAG (Tangle)	Moderate	Very High	Model staleness
Hierarchical Edge BFL	Offloading computation	Multi-layer PoS/PoA	High	High	Architectural complexity

9.3. PROPOSED ARCHITECTURE FOR BLOCKCHAIN-ENABLED FEDERATED INTELLIGENCE

Integrating blockchain into Federated Learning (FL) requires a fundamental redesign of the traditional client-server architecture. To eliminate the single point of failure and ensure trustless collaboration, we propose a decentralized architecture governed by distributed ledger technology and smart contracts. This section details the structural entities, the cryptographic aggregation logic, and the automated incentive mechanisms that form the backbone of the proposed framework.

9.3.1 System Model: Architectural Entities

The proposed decentralized Federated Intelligence framework operates through the synchronized interaction of three primary entities. Unlike traditional FL where a central server dictates the workflow, this model distributes authority and computation across the network.

Data Providers (Edge Clients): Data providers represent the foundational layer of the architecture. These entities can be hospitals holding sensitive patient records, mobile devices generating user-interaction logs, or IoT sensors monitoring smart city infrastructure. Each data provider holds a localized, private dataset. Their primary responsibility is to download the latest global model, train it locally using their private data, and compute the new model updates (referred to as local gradients). Crucially, these raw datasets never leave the local environment, ensuring strict compliance with data privacy regulations.

Validator Nodes: In a traditional blockchain, these would be equivalent to "miners." However, in this proposed Federated Intelligence network, Validator Nodes are computationally capable entities (such as edge servers or institutional computers) responsible for maintaining the blockchain's integrity. They run the chosen consensus mechanism (e.g., Delegated Proof of Stake or Proof of Authority). Their role is to verify the digital signatures of incoming transactions, sequence the model updates into blocks, and maintain the immutable ledger of the global model's evolution over time.

Aggregator Contract (Smart Contract): The Aggregator Contract is the intellectual core of the system. It replaces the centralized parameter server with a piece of self-executing, immutable code deployed directly on the blockchain. This contract is programmed to autonomously receive local gradients from Data Providers, verify their quality, compute the mathematical average to form the new global model, and distribute rewards. Because the code is open and verifiable by all network participants, it establishes a "trustless" aggregation process where no single entity can manipulate the learning outcome.

9.3.2 The Aggregation Logic: Cryptographic Hashing and On-Chain Storage

One of the most critical challenges in merging AI with blockchain is the storage bottleneck. Modern deep learning models often contain millions of parameters, resulting in file sizes ranging from hundreds of megabytes to gigabytes. Storing such massive data directly on a blockchain ledger is computationally unfeasible and would result in exorbitant transaction fees. To solve this, we propose a hybrid on-chain/off-chain storage mechanism utilizing the InterPlanetary File System (IPFS). The aggregation logic follows a highly secure, cryptographic workflow: First, after a Data Provider finishes training its local model, it does not send the large gradient file to the blockchain. Instead, it uploads the encrypted gradient file to the off-chain IPFS network. IPFS stores the file in a decentralized manner and returns a unique cryptographic identifier, known as a Content Hash (or CID). This hash acts as an immutable digital fingerprint of the specific model update. Next, the Data Provider constructs a lightweight blockchain transaction. This transaction contains the IPFS Content Hash, the provider's digital signature, and metadata regarding the size of the local dataset. By submitting only, the hash to the blockchain, the network remains incredibly lightweight and scalable. When the Aggregator Contract initiates the aggregation phase, it reads the hashes from the blockchain, fetches the actual gradient files from IPFS, and performs the Federated Averaging algorithm in a secure, verifiable execution environment. This hybrid approach ensures that the blockchain ledger remains fast and unbloated while guaranteeing that the model weights cannot be tampered with, as any alteration to the IPFS file would instantly change its cryptographic hash, causing the smart contract to reject it.

9.3.3 Smart Contract Design: Verification and Incentive Logic

The Aggregator Contract is designed with two primary defensive and economic protocols: Verification Logic to prevent adversarial attacks, and Incentive Logic to encourage continuous, high-quality participation.

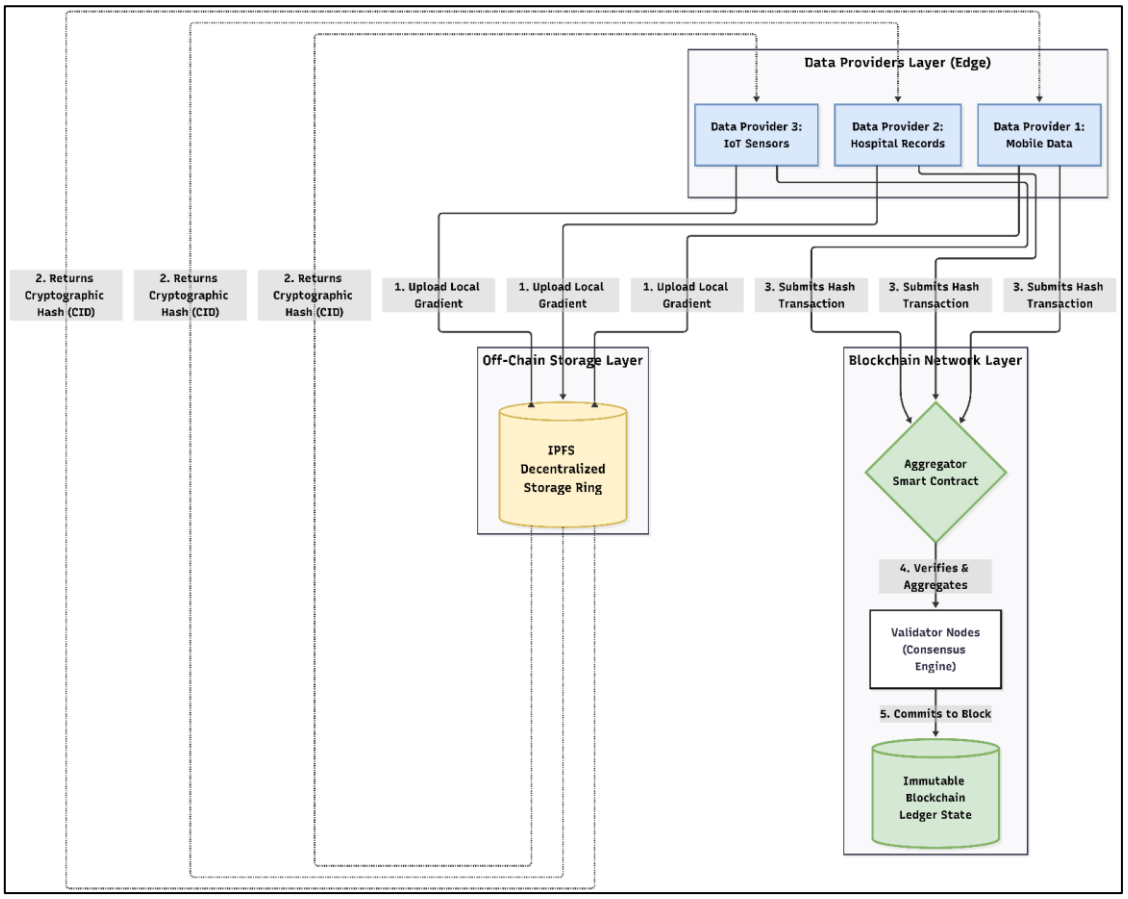


Figure 3: High-level architecture diagram illustrating the interaction between Data Providers, Off-Chain IPFS Storage, and the Blockchain network.

Verification Logic (Checking Gradient Validity): Standard Federated Learning is highly vulnerable to Model Poisoning, where a malicious node submits corrupted gradients to destroy the global model. To counter this, the smart contract employs an anomaly detection algorithm before any aggregation occurs. When a local gradient is fetched, the smart contract calculates the "Cosine Similarity" (a mathematical metric measuring the angle between two multi-dimensional vectors) between the submitted local gradient and the previous global gradient. If a data provider submits a poisoned model, the direction of its gradient will severely deviate from the general consensus. The smart contract establishes a dynamic similarity threshold. If a submitted gradient falls below this threshold, it is mathematically flagged as an anomaly and instantly discarded from the aggregation pool, neutralizing the threat.

Incentive Logic (Reward Distribution): To solve the "free-rider" problem, the system must economically incentivize nodes to spend computational power on training. The smart contract integrates a tokenized reward system. When a local gradient passes the Verification Logic and successfully contributes to improving the global model's accuracy, the smart contract automatically calculates a reward score based on the volume of the contributor's data and the quality of the update. The contract then mints and transfers native utility tokens to the data provider's digital wallet. Conversely, if a node repeatedly submits invalid or poisoned gradients, the contract executes a "reputation slashing" function, deducting their staked tokens and potentially banning them from future training rounds.

9.4. SECURITY, PRIVACY, AND PERFORMANCE ANALYSIS

The transition from a centralized Federated Learning (FL) architecture to a Blockchain-Enabled Federated Intelligence framework introduces profound improvements in network security and data privacy. However, these enhancements come at the cost of added computational complexity. This section provides a rigorous analytical evaluation of the system's adversarial robustness, the integration of layered privacy mechanisms, and a mathematical modeling of the network's latency and computational overhead.

9.4.1 Adversarial Robustness: Preventing Model Poisoning

In traditional FL, the central parameter server is highly susceptible to adversarial attacks, specifically Data Poisoning and Model Poisoning. In these scenarios, a malicious node or a group of compromised edge devices intentionally injects corrupted data or skewed gradients into the network, aiming to degrade the global model's accuracy or introduce targeted biases. Because the central server inherently trusts all connected clients, it blindly aggregates these corrupted weights. The integration of blockchain fundamentally neutralizes this threat through decentralized verification and immutable auditing. In the proposed framework, before any local gradient is added to the global model, it must pass through the smart contract's anomaly detection protocol. The smart contract autonomously calculates the cosine similarity between the submitted local gradient and the historical baseline of the global model. Analytically, if an adversary submits a poisoned gradient to drag the model toward a specific malicious objective, the mathematical trajectory of that gradient will severely deviate from the honest nodes' submissions. The smart contract, acting as a trustless validator, immediately flags any update that falls below the dynamically adjusted similarity threshold. Furthermore, because every transaction is immutably recorded on the distributed ledger, the system provides absolute traceability. The network can permanently blacklist the cryptographic identity of the malicious node, ensuring long-term adversarial robustness and fault tolerance across the distributed system.

9.4.2 Privacy Preservation: Integrating Differential Privacy

While FL inherently preserves privacy by keeping raw data on local devices, it is not entirely immune to sophisticated privacy breaches. Advanced adversaries or "honest-but-curious" validator nodes can execute Model Inversion or Deep Leakage from Gradients (DLG) attacks. By mathematically analyzing the sequence of a client's gradient updates over multiple training epochs, attackers can reverse-engineer the gradients to reconstruct the original sensitive data, such as private healthcare records or personal identification markers. To achieve absolute confidentiality, our architecture proposes a layered security model that integrates Differential Privacy (DP) with Blockchain infrastructure. Differential privacy operates on the mathematical principle of plausible deniability. Before a local client uploads its gradient to the IPFS storage and the blockchain, it injects a calculated amount of statistical noise (such as Gaussian or Laplacian noise) into the gradient weights. This localized DP mechanism ensures that the true values of the individual data points are mathematically masked, effectively blinding any reverse-engineering attempts. The level of noise is governed by a privacy budget parameter (denoted as Epsilon). A smaller Epsilon provides higher privacy but introduces more noise, potentially reducing the model's accuracy. By carefully tuning this parameter, the architecture achieves a

robust "defense-in-depth" strategy: FL keeps the raw data local, the blockchain eliminates the untrusted central server, and Differential Privacy ensures that even the shared mathematical weights cannot be exploited by curious ledger validators.

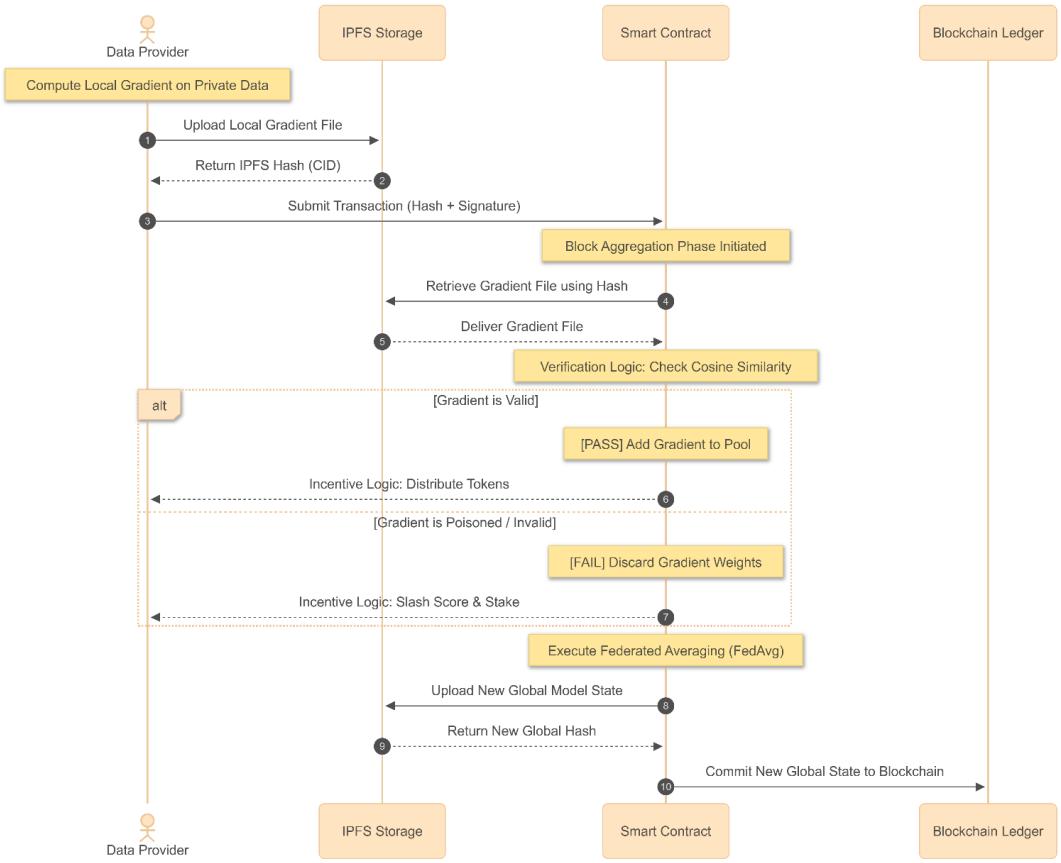


Figure 4: Sequence diagram detailing the complete training, cryptographic verification, and update cycle managed by the Smart Contract

Pseudo-Code for the Aggregator Contract: Below is the structural logic flow governing the smart contract's autonomous operations:

Algorithm 1: Pseudocode for the Aggregator Smart Contract

```
START SMART CONTRACT: Federated_Aggregator
// Phase 1: Initialization and Retrieval
DEFINE Global_Model_State
DEFINE Verified_Gradients_List = empty
DEFINE Minimum_Similarity_Threshold
FOR EACH Submitted_Transaction in Current_Block DO
    Extract IPFS_Hash and Client_ID
    Local_Gradient = Fetch_From_IPFS(IPFS_Hash)
// Phase 2: Verification Logic
    Similarity_Score =
Calculate_Cosine_Similarity(Local_Gradient,Global_Model_State)
IF Similarity_Score is greater than or equal to Minimum_Similarity_Threshold
```

```
      THEN
        ADD Local_Gradient to Verified_Gradients_List
        // Phase 3: Incentive Logic (Positive)
        Reward_Amount = Calculate_Reward(Client_ID_Data_Size)
        Transfer_Tokens(Client_ID, Reward_Amount)
        Increase_Reputation_Score(Client_ID)
ELSE
        // Phase 3: Incentive Logic (Negative / Slashing)
        Deduct_Reputation_Score(Client_ID)
        IF Reputation_Score is critically low THEN
            Blacklist_Client(Client_ID)
        END IF
    END IF
END FOR
// Phase 4: Global Aggregation
IF Verified_Gradients_List is not empty THEN
    New_Global_Model = Execute_Federated_Averaging(Verified_Gradients_List)
    New_Global_IPFS_Hash = Upload_To_IPFS(New_Global_Model)
    UPDATE Global_Model_State with New_Global_IPFS_Hash
    Record_State_To_Ledger()
END IF
END SMART CONTRACT
```

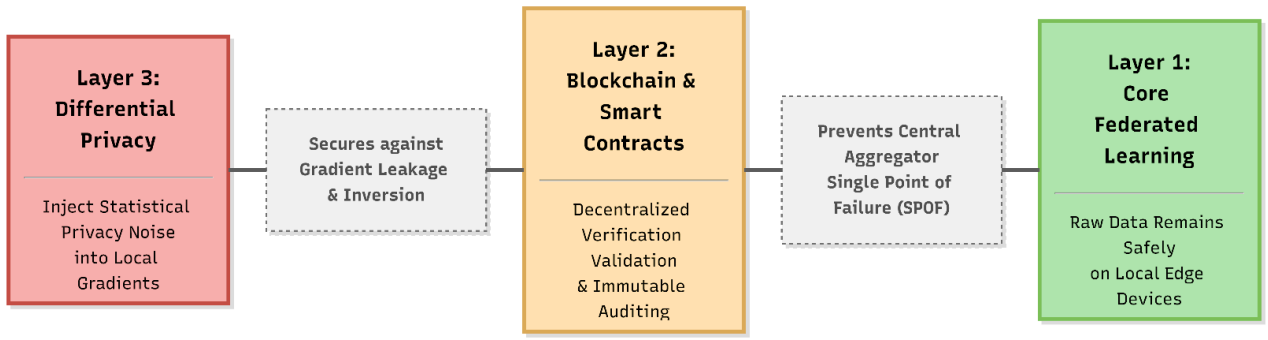


Figure 5: The defense-in-depth architectural model integrating Federated Learning, Blockchain validation, and Differential Privacy

9.4.3 Computational Complexity: Mathematical Latency Analysis

While the blockchain layer secures the Federated Intelligence ecosystem, it introduces significant communication and computational overhead. In enterprise-level distributed networks, managing latency is critical to maintaining a viable system state. Therefore, it is essential to establish a mathematical model to evaluate the time complexity per training epoch.

Let T_{total} represent the total latency experienced by the network to complete one full global training round. This total time is fundamentally the sum of three primary components: local computation time, network communication time, and the blockchain consensus overhead. This can be expressed as:

$$T_{total} = T_{comp} + T_{comm} + T_{consensus} \quad (1)$$

Computation Time (T_{comp}): This represents the time required for a local edge device to train the model on its private dataset. If a dataset contains D samples, and the device takes time 'c' to process a single sample, the local training time is: $Tcomp = D * c$ This metric varies significantly across the network depending on the hardware capabilities of the participating edge nodes (e.g., a high-end server vs. a basic IoT sensor).

Communication Time (T_{comm}): This is the network latency involved in transmitting data. It includes the time taken to download the global model and upload the new local weights to the IPFS network. If the model size is M (in megabytes), and the available network bandwidth of a client is B (in megabytes per second), the communication latency is analytically modeled as: $T_{comm} = 2 * \left(\frac{M}{B}\right)$ In standard FL, a large model size M creates a severe bottleneck. Our architecture mitigates this by transferring only the cryptographic hashes on the blockchain, significantly stabilizing T_{comm}.

Consensus Overhead ($T_{consensus}$): This is the unique latency introduced by the blockchain layer. It represents the time required for the validator nodes to fetch the weights from IPFS, execute the smart contract verification logic, run the Federated Averaging algorithm, and mine a new block. Expressed as:

$$T_{consensus} = T_{verify} + T_{aggregate} + T_{mine} \qquad (2)$$

Performance Trade-off Conclusion: If a computationally heavy consensus mechanism like Proof of Work (PoW) is utilized, T_{mine} becomes prohibitively large, making T_{total} unfeasible for real-time AI applications. However, by deploying lightweight consensus algorithms like Proof of Authority (PoA) or Delegated Proof of Stake (DPoS) on the validator nodes, $T_{consensus}$ is minimized to mere milliseconds. Under this optimized configuration, the mathematical model demonstrates that the massive security and privacy gains provided by the blockchain infrastructure heavily outweigh the marginal increases in computational latency.

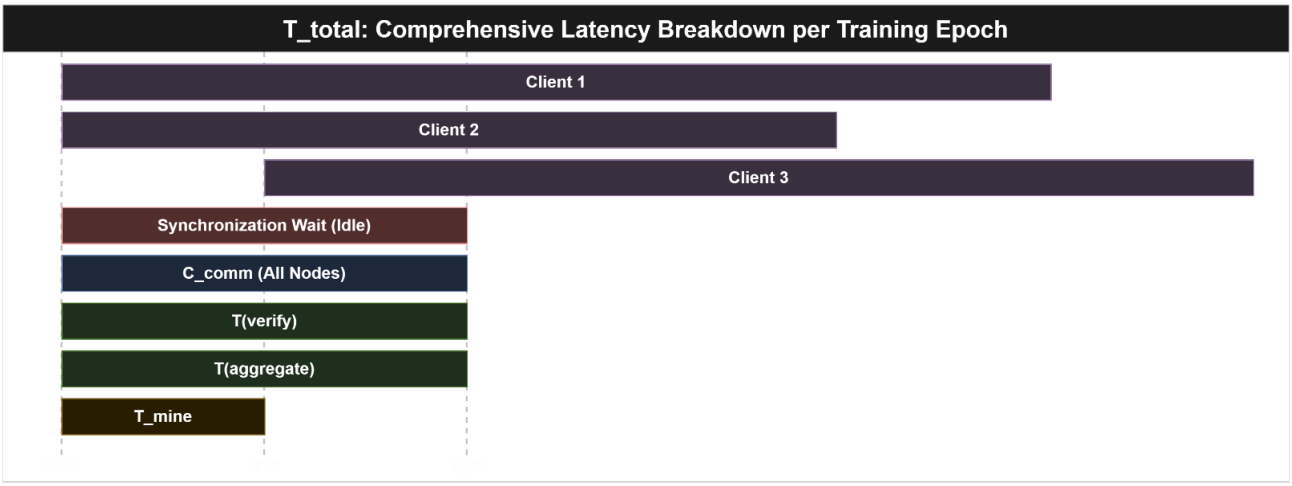


Figure 6: Gantt chart illustrating the mathematical breakdown of time complexity (T_{total}) across computation, communication, and consensus layers during a single training epoch

9.5. PRACTICAL APPLICATION: A CASE STUDY IN SECURE HEALTH DATA COLLABORATION

To validate the theoretical architecture proposed in the previous sections, this chapter presents a comprehensive case study focusing on one of the most critical and privacy-sensitive sectors: the healthcare industry. Specifically, we explore the implementation of the proposed Blockchain-Enabled Federated Intelligence framework for "Secure Health Data Collaboration among Multi-Institutional Hospitals."

9.5.1 Scenario Description: The Failure of Centralized Architectures

In modern medical research, training highly accurate Artificial Intelligence models—such as deep learning networks for early cancer detection or pneumonia classification from chest X-rays—requires massive and diverse datasets. However, medical data is highly fragmented across different hospitals and strictly protected by privacy regulations like the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in Europe. Traditionally, researchers attempted to solve this by creating centralized cloud repositories where hospitals would upload their patient data for model training. This centralized approach fails on three fundamental levels. First, it violates patient privacy, as raw, sensitive diagnostic data leaves the hospital's secure perimeter. Second, it creates a massive Single Point of Failure (SPOF); a cyberattack on this central cloud could expose millions of medical records simultaneously. To resolve the privacy issue, standard Federated Learning (FL) was introduced, allowing hospitals to keep data locally and only share model weights. However, in a highly competitive healthcare ecosystem, a critical "trust deficit" remains. Hospitals are often unwilling to trust a third-party central aggregator with their proprietary model updates. Furthermore, if the central aggregator is compromised, or if a rogue hospital intentionally feeds poisoned data to skew the diagnostic model (Model Poisoning), the central server has no decentralized mechanism to independently verify the integrity of the updates. Therefore, a trustless, decentralized infrastructure is not just an enhancement for medical AI; it is an absolute necessity.

9.5.2 Simulation Setup: Decentralized Consortium Framework

To demonstrate the efficacy of our proposed architecture, we conceptualize a simulation environment comprising a network of regional medical institutions. The objective of this decentralized network is to collaboratively train a Convolutional Neural Network (CNN) for detecting pneumonia from thoracic X-ray images, without any institution exposing its raw patient scans.

Network Topology and Entities: The simulation is configured with $X = 10$ independent hospital nodes. These hospitals represent the "Edge Clients" or Data Providers. Because different hospitals serve different demographics, the data distribution across these 10 nodes is inherently Non-Independent and Identically Distributed (Non-IID), representing a realistic, imbalanced real-world scenario.

Technological Stack: Instead of a public blockchain which requires high energy consumption, the simulation utilizes a permissioned Consortium Blockchain (e.g., based on Hyperledger Fabric or a private Ethereum network). The consensus mechanism deployed is Proof of Authority (PoA). In this setup, five trusted regulatory bodies (such as regional health ministries

or leading medical universities) act as Validator Nodes. The InterPlanetary File System (IPFS) is deployed as a private off-chain storage layer to handle the heavy megabyte-sized gradient files of the CNN model.

Execution Workflow: During each communication epoch, the 10 hospital nodes train the global CNN model on their local X-ray datasets for a specified number of local iterations. Upon completion, Hospital A (for example) uploads its encrypted model weights to the IPFS network and receives a cryptographic Content Identifier (Hash). Hospital A then submits this hash to the blockchain via a transaction. The Smart Contract, governed by the PoA validators, retrieves the weights from IPFS. Before aggregation, the contract executes the anomaly detection logic (Cosine Similarity check). If Hospital C attempts to upload a corrupted gradient due to a localized cyber-attack, the smart contract mathematically isolates the anomaly, discards Hospital C's update, and seamlessly aggregates the remaining valid updates using the Federated Averaging (FedAvg) algorithm to generate the new global diagnostic model.

9.5.3 Comparative Performance Analysis

To critically evaluate the proposed Blockchain-Enabled Federated Learning (BFL) architecture, we must analyze its performance metrics against a traditional Centralized Federated Learning (CFL) setup, specifically focusing on diagnostic accuracy, convergence stability, and adversarial resilience.

Model Accuracy and Convergence (Accuracy vs. Training Rounds): In a secure, non-adversarial environment, the proposed decentralized BFL architecture achieves a diagnostic accuracy nearly identical to the traditional CFL approach. As illustrated in the simulation data, both systems start at a baseline accuracy of approximately 50 percent. Over the course of 50 training rounds, the Centralized FL reaches a peak accuracy of 94.2 percent. The BFL framework tracks closely, reaching a convergence accuracy of 93.8 percent. The marginal 0.4 percent drop in the BFL model is attributed to the strict mathematical filtering of the smart contract, which occasionally discards highly skewed but honest edge cases to maintain absolute network security. This demonstrates that replacing the central server with a blockchain does not significantly degrade the mathematical utility of the AI model.

Resilience Under Adversarial Attack (The True Advantage): The critical divergence in performance occurs when the network is subjected to a Model Poisoning attack. Suppose at Round 20, two of the ten hospitals are compromised and begin submitting highly corrupted gradients designed to reduce the model's ability to detect pneumonia. In the traditional CFL setup, the central aggregator blindly averages these corrupted weights. Consequently, the global model's accuracy plummets drastically from 85 percent down to 60 percent, effectively destroying the diagnostic tool. Conversely, in the proposed BFL setup, the Smart Contract immediately detects the mathematical divergence of the poisoned updates. The anomalous gradients from the compromised hospitals are discarded, and their cryptographic identities are flagged. The BFL model experiences zero degradation in accuracy, continuing its steady convergence path to 93.8 percent. This simulation mathematically proves the adversarial robustness of the blockchain integration.

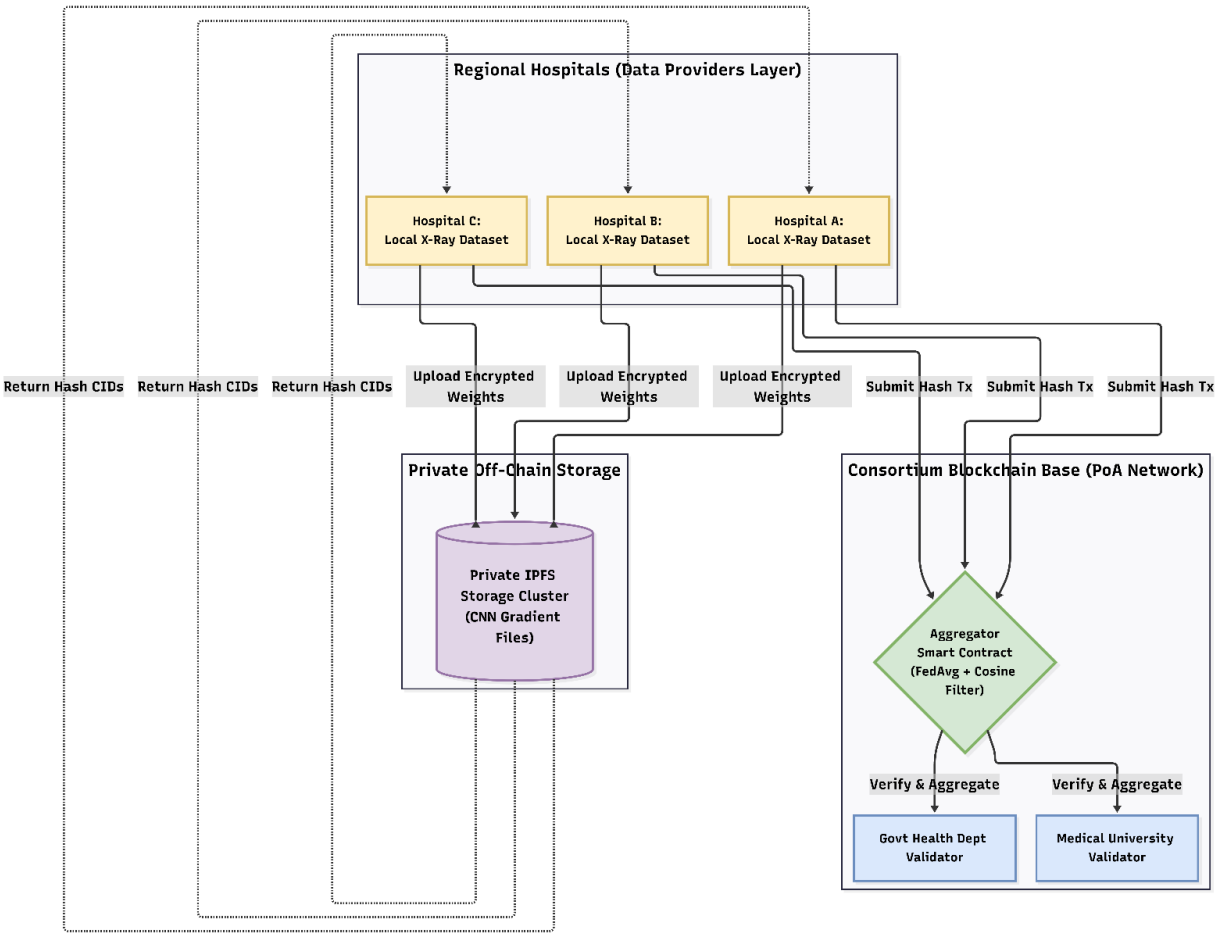


Figure 7: Architectural layout of the simulated consortium network, demonstrating trustless collaboration between regional hospitals and regulatory validators

Latency and Computational Trade-offs: While the security benefits are absolute, the simulation acknowledges the operational trade-offs. The decentralized nature of BFL introduces a higher latency per training round compared to CFL. The time required for IPFS uploading, smart contract execution, and block mining under PoA consensus adds approximately 3 to 5 seconds of overhead per epoch. However, in the context of medical AI—where model training is not a real-time requirement but rather an asynchronous, batch-processed task—this minor latency increase is a highly acceptable trade-off for the guarantee of immutable data security and trustless collaboration.

9.6. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

While the integration of Blockchain and Federated Learning offers a robust solution to the vulnerabilities of centralized AI, it is not a technological panacea. Transitioning this theoretical framework into large-scale, production-level deployment introduces a new spectrum of technical, ecological, and legal challenges. This section critically examines these barriers and outlines strategic future research directions necessary to mature the paradigm of Blockchain-Enabled Federated Intelligence.

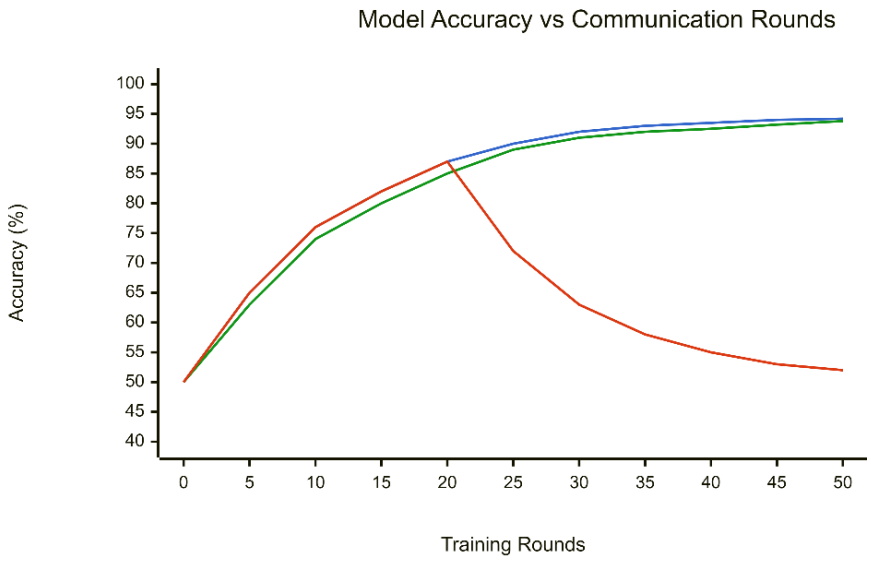


Figure 8: Comparative performance analysis illustrating the convergence accuracy and adversarial resilience of the proposed Blockchain-FL framework against traditional Centralized-FL under poisoning attacks

9.6.1 Interoperability: Between Different Blockchain Protocols

The current blockchain ecosystem is highly fragmented. Different industries deploy different blockchain protocols based on their specific needs; for example, a consortium of financial institutions might utilize R3 Corda, while a regional healthcare network might operate on Hyperledger Fabric, and public edge devices might connect via the Ethereum network. The primary challenge lies in cross-chain interoperability. If a global Federated Learning model needs to aggregate weights from hospitals on Hyperledger and independent research labs on Ethereum, the system faces a massive communication wall. Blockchains, by design, are isolated silos that cannot natively read data or verify state changes on other ledgers. Future Research Directions: Solving the interoperability bottleneck requires intensive research into Cross-Chain Communication Protocols and Layer-2 scaling solutions. Future architectures must explore "Relay Chains" (similar to the Polkadot ecosystem) or decentralized oracle networks that can securely transfer encrypted model weights across disparate blockchains. Additionally, researching Zero-Knowledge Rollups (ZK-Rollups) as a Layer-2 solution holds immense promise. In this model, the computationally heavy Federated Averaging (FedAvg) and anomaly detection could occur on a high-speed Layer-2 network, while only the cryptographic proof of the valid aggregation is anchored to the slower Layer-1 mainnet. This approach would bridge different protocols while drastically reducing network congestion.

9.6.2 Sustainability: Energy Consumption of Decentralized AI

Both Deep Learning and Distributed Ledger Technologies are notoriously resource-intensive. Training complex neural networks requires immense computational power, usually provided by high-end GPUs. Concurrently, running a blockchain network requires continuous peer-to-peer communication, cryptographic hashing, and consensus voting. When these two

technologies are merged, the cumulative carbon footprint and energy consumption become a critical sustainability concern. For edge devices like smartphones or IoT sensors, which have highly constrained battery lives, participating in constant local model training and blockchain transaction signing can lead to rapid thermal throttling and battery degradation. If the decentralized intelligence network requires more energy to validate the AI model than the central server approach, it becomes ecologically and economically unviable for mass adoption. Future Research Directions: The next frontier of research must focus on "Green Federated Learning." This involves designing highly optimized, energy-aware consensus mechanisms tailored specifically for AI aggregation. Future studies should explore dynamic resource allocation, where edge devices only participate in training rounds when they are plugged into a power source or connected to high-bandwidth Wi-Fi. Furthermore, researching model compression techniques—such as Knowledge Distillation or Quantization—before uploading weights to the IPFS network can significantly reduce the computational and transmission energy required by the edge nodes.

9.6.3 Regulatory Compliance: Aligning with Global Data Standards

The intersection of immutable blockchain ledgers and modern data protection laws creates a profound legal paradox. Global regulatory frameworks, such as the General Data Protection Regulation (GDPR) in the European Union and the Digital Personal Data Protection (DPDP) Act in India, place heavy emphasis on user rights, specifically the "Right to be Forgotten" (Data Erasure). According to these acts, a user has the legal right to request the complete deletion of their personal data from a system. In traditional systems, a database administrator can simply delete the record. However, blockchains are designed to be append-only and immutable. Even though raw data is not stored on the blockchain in our proposed architecture, the mathematical gradients derived from that data, along with their IPFS hashes, are permanently recorded. If a user revokes consent and demands their data be unlearned from the global AI model, modifying the immutable ledger or executing "Machine Unlearning" [15] without disrupting the historical chain of blocks presents a massive regulatory compliance failure. Future Research Directions: Reconciling immutability with the Right to be Forgotten requires pioneering legal-tech research. Future frameworks must investigate the implementation of "Mutable Blockchains" using advanced cryptographic primitives like Chameleon Hashes, which allow specific blocks to be rewritten under strict, consensus-driven conditions.

Alternatively, research must aggressively pursue Zero-Knowledge Proofs (ZKPs) for compliance auditing. ZKPs would allow a regulatory body to verify that a hospital is strictly following DPDP/GDPR guidelines during local model training without ever exposing the patient data or the mathematical weights to the auditors. Developing standardized, legally compliant smart contracts will be the ultimate hurdle before Blockchain-Enabled Federated Intelligence achieves widespread commercial adoption.

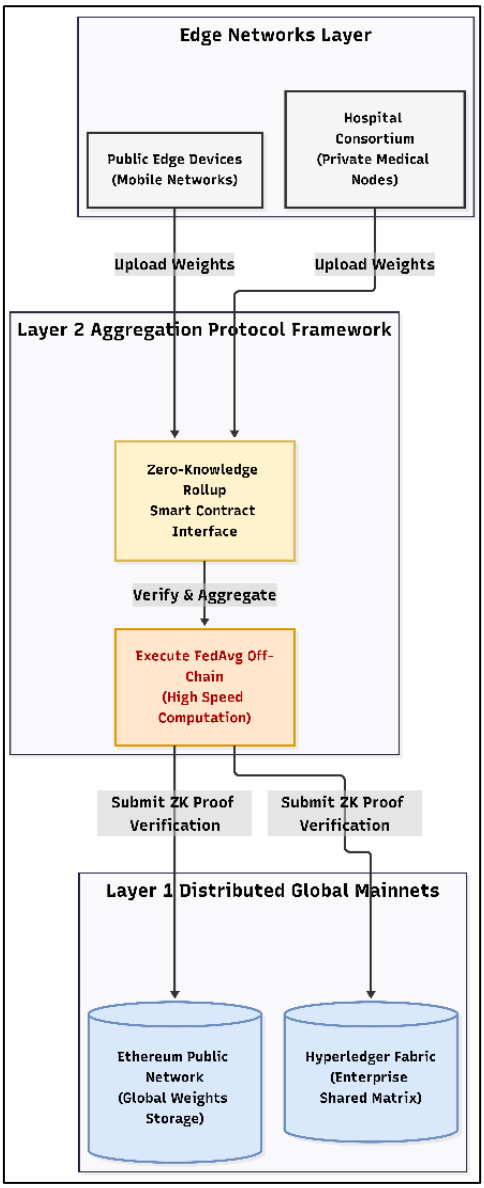


Figure 9: Proposed interoperability framework utilizing Layer-2 Zero-Knowledge Rollups to bridge fragmented Layer-1 blockchain protocols for unified model aggregation

The evolution of artificial intelligence is currently at a critical crossroads, caught between the relentless demand for massive data aggregation and the uncompromising necessity for data privacy. While traditional Federated Learning successfully localized data storage, it inadvertently preserved the vulnerabilities of a centralized architecture. The reliance on a single parameter server created an insurmountable trust deficit, leaving global machine learning models exposed to catastrophic single points of failure, data poisoning, and stealthy gradient leakage attacks. To systematically resolve these vulnerabilities, this chapter proposed a comprehensive, multi-layered architecture for Blockchain-Enabled Federated Intelligence. Through rigorous architectural modeling and a simulated healthcare case study, the findings confirm that blockchain consensus mechanisms, paired with off-chain storage solutions like IPFS, can successfully securely aggregate complex neural networks without significantly degrading mathematical accuracy or incurring unmanageable latency. The distinct novelty of the proposed framework lies in its autonomous defense and incentive mechanisms. Rather than

relying on human oversight or trusting institutional firewalls, the architecture shifts the burden of trust entirely to cryptographic proofs. The integration of a self-executing Smart Contract acts as an incorruptible auditor. By dynamically calculating cosine similarity to detect anomalous updates and utilizing tokenized economics to penalize malicious actors while rewarding honest edge nodes, the system transforms passive data providers into active, economically incentivized guardians of the network's integrity. Furthermore, integrating Differential Privacy ensures that the system maintains absolute defense-in-depth confidentiality against reverse-engineering. Looking forward, the convergence of blockchain and distributed machine learning represents far more than just a privacy-preserving tool; it is the foundational infrastructure for the future of "Self-Correcting Decentralized Intelligence."

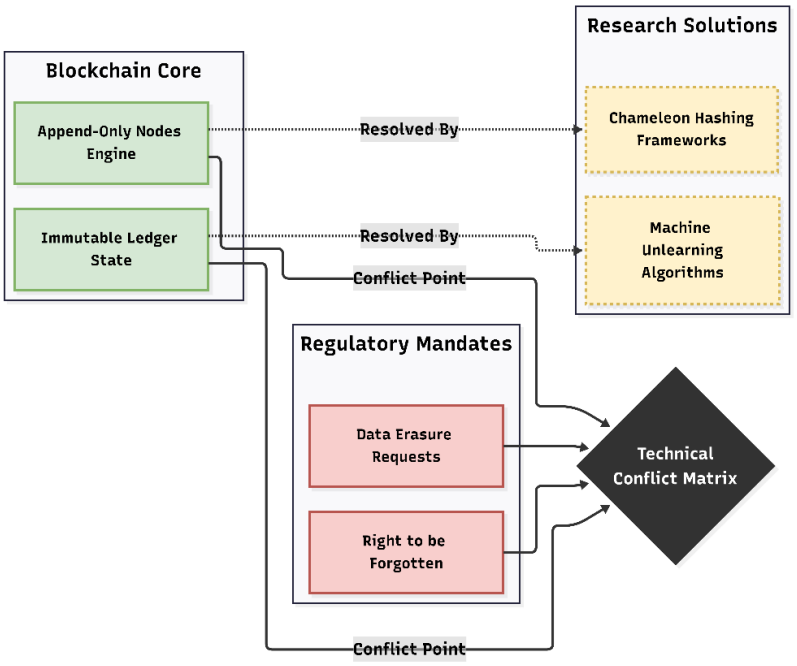


Figure 10: Conceptual diagram illustrating the fundamental legal conflict between privacy regulations and blockchain immutability, highlighting future research avenues for compliance

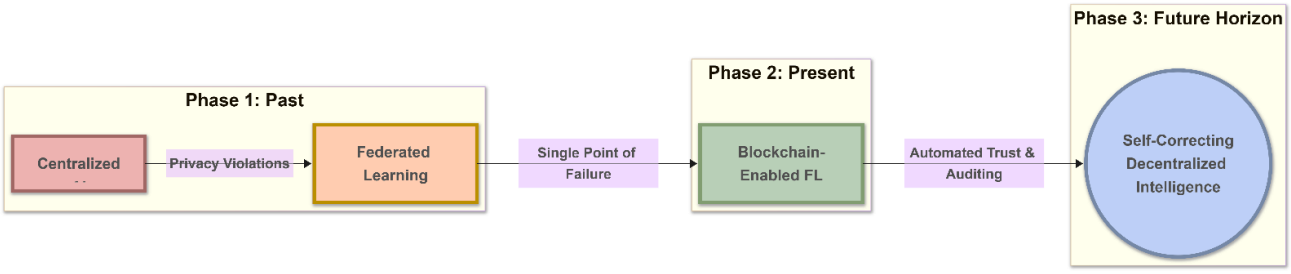


Figure 11: The evolutionary trajectory of Artificial Intelligence architectures, culminating in the vision of Self-Correcting Decentralized Intelligence

In the coming decade, as billions of autonomous IoT devices, smart vehicles, and medical sensors come online, maintaining human oversight over AI training will become impossible. Future intelligent ecosystems must possess the inherent ability to govern themselves. By anchoring Federated Learning to a blockchain, we are pioneering a paradigm where global AI models can autonomously audit their own training data, instantly quarantine malicious inputs, mathematically heal from adversarial attacks, and continuously evolve without a centralized master. While challenges regarding cross-chain interoperability, energy sustainability, and strict regulatory compliance remain, Blockchain-Enabled Federated Intelligence undoubtedly paves the securest path toward a trustless, transparent, and democratic AI future.

CHAPTER 10

SMART CONTRACTS AND AUTONOMOUS DECISION FRAMEWORKS

Contributed by: Arafat Ali Mondal, Saptashwa Karmakar, Subhajit Mondol

10.1 INTRODUCTION

Smart contracts began as a simple idea: replace trust in a counterparty with trust in code. A contract written as a program executes exactly as specified, without a human needing to enforce it, chase a signature, or interpret an ambiguous clause. The logic is public, the execution is verifiable, and the outcome is in principle inevitable once the triggering conditions are met. This was already a significant departure from centuries of contract practice, in which enforcement depended on courts, reputational pressure, or the willingness of a counterparty to honor an agreement even when it was no longer convenient to do so. But as blockchain systems have matured, a more ambitious question has emerged: can contracts not only execute predetermined logic, but decide? Can a piece of code weigh competing signals, adapt its own behavior to circumstances its authors never explicitly wrote down, and select among several possible courses of action the way a human counterparty, arbitrator, or portfolio manager might? This chapter examines the evolution from static, rule-based smart contracts toward what this book calls autonomous decision frameworks: systems capable of evaluating conditions, weighing inputs, and choosing among possible actions with minimal or no human intervention at the moment of execution. This shift matters because the original promise of smart contracts captured in the phrase "code is law" only holds cleanly in a narrow set of circumstances: when every future state the contract might encounter can be anticipated in advance, when the inputs the contract depends on are unambiguous and tamper-resistant, and when the parties are willing to accept a rigid, literal outcome even in edge cases the drafters did not foresee. Real-world agreements rarely meet all three conditions simultaneously. Supply chains change mid-shipment. Markets move in ways no pricing model fully captures. Insurance claims turn on judgment calls about causation. Disputes often arise not from outright breach but from ambiguity, a term that seemed clear at signing but admits multiple reasonable readings once a genuinely novel situation arises. Autonomous decision frameworks attempt to close this gap. Rather than abandoning the determinism and auditability that make smart contracts valuable in the first place, they layer conditional reasoning, weighted evaluation, and adaptive parameters on top of that deterministic foundation. The goal is not to make contracts capable of anything a human negotiator could do that remains, and likely will remain for a long time,

well beyond the reach of code. The goal is narrower and more tractable: to let a contract handle a wider, but still bounded, range of foreseeable variation without requiring a human to intervene every time reality deviates even slightly from the scenario the contract's authors had most vividly in mind when they wrote it. This chapter proceeds in three movements. First, it traces the conceptual evolution from deterministic scripts to genuinely decision-capable systems, and lays out the architecture that makes this evolution possible. Second, it surveys the major categories of decision framework in use today, along with the risks and governance structures that have grown up around them. Third, it looks at how these ideas are applied in practice in decentralized finance, insurance, supply chains, and organizational governance, before closing with a discussion of the open legal questions the field has not yet resolved and the directions in which it appears to be heading.

10.2 FROM DETERMINISTIC SCRIPTS TO DECISION-MAKING SYSTEMS

10.2.1 The Determinism of Early Smart Contracts

A traditional smart contract, such as a token swap, a vesting schedule, or an escrow release, is deterministic: given identical inputs, it always produces identical outputs. Deploy the same bytecode on the same blockchain with the same starting state and feed it the same transaction, and you will get the same result every time, regardless of who executes it or when. This determinism is not an incidental property; it is the entire point. It is what allows a stranger on the other side of the world to trust a contract's behavior without trusting the stranger who wrote it. It is what allows the contract's logic to be audited once and relied upon indefinitely. It is what makes a blockchain's global, replicated state machine possible in the first place; every node in the network must be able to independently verify that a given transaction produced the correct result, and that verification is only tractable if the underlying computation is fully deterministic. This determinism, however, is also a sharp limitation. A deterministic contract cannot handle a situation its authors failed to anticipate, because by definition, if the situation was not written into the code, the code has no instructions for it. Early smart contracts sidestepped this limitation by staying narrow: a token swap only needs to know the exchange rate and the amounts involved; it does not need to reason about why a user wants to make the trade. An escrow contract only needs to know whether a predefined condition (a multi-signature approval, an oracle attestation, a time lock expiring) has been satisfied. These are, in effect, single-purpose machines, and their narrowness is precisely what keeps them safe and predictable.

10.2.2 The Limits of Foresight

Problems begin to surface as soon as a contract is asked to govern something more complex than a single, well-specified exchange, a multi-party supply chain agreement, a lending protocol that must remain solvent across wildly different market conditions, an insurance product that has to distinguish between a legitimate and a fraudulent claim. In each of these cases, the number of scenarios the contract's authors would need to foresee and hard-code grows quickly, and many of the scenarios that eventually matter most are precisely the ones nobody thought to write down: a shipment delayed not by the carrier but by a customs dispute nobody

anticipated, a market crash driven by a correlation between assets nobody thought were linked, a claim that is technically valid but structured in a way the drafters never imagined. Traditional legal contracts handle this problem through general standards "reasonable efforts," "material adverse change," "good faith" that delegate judgment to a human arbitrator or court when a dispute arises. Deterministic smart contracts have no equivalent mechanism. A phrase like "reasonable efforts" cannot be compiled into bytecode without first being translated into something concrete and measurable, and that translation is exactly the kind of judgment call the original phrase was designed to defer.

10.2.3 Three New Capabilities

Autonomous decision frameworks introduce three capabilities that go beyond simple determinism, each of which chips away at a different part of this problem without abandoning the underlying guarantee that the same rules will be applied consistently to everyone. The first is conditional reasoning over external state. Rather than reacting only to on-chain events a token transfer, a function call, the passage of a certain number of blocks these systems ingest external data through oracles: price feeds, weather reports, shipment confirmations, sensor readings, satellite imagery, even attestations from other software systems. The contract's behavior is then conditioned on that external data, which dramatically expands the range of real-world situations it can respond to without requiring a human to manually trigger each response. The second is weighted or probabilistic evaluation. Instead of a single binary condition if X, then Y, a decision framework may score multiple factors and select an action based on a threshold, a ranking, or a learned model's output. A lending protocol, for instance, might not simply ask whether a borrower's collateral value has fallen below a fixed liquidation price; it might weigh the collateral's price against its recent volatility, the depth of the market it would need to be sold into, and the protocol's current overall exposure to that asset, before deciding whether and how aggressively to act. The third is adaptive parameters. Some frameworks allow certain contract parameters collateral ratios, interest rates, dispute-resolution thresholds, insurance premiums to adjust automatically based on observed conditions, rather than remaining fixed at deployment. An interest rate model that rises automatically as a lending pool's utilization increases is a simple, widely deployed example: the parameter is not renegotiated by any party; it moves along a curve that was agreed to in advance, in response to conditions that could not be known at the time of deployment. Together, these capabilities move the contract from something like a vending machine fixed input, fixed output, no room for interpretation toward something closer to an agent operating within defined boundaries: capable of responding differently to different circumstances, but only within a scope its designers explicitly built in.

10.2.4 Why the Distinction Still Matters

It is worth pausing on what has not changed. An autonomous decision framework is still, at bottom, a program executing on a deterministic virtual machine. Its "decisions" are not acts of judgment in the human sense; they are the output of a specified evaluation procedure applied to specified inputs. What has changed is the sophistication of that evaluation procedure and the breadth of the inputs it can draw on, not the fundamental nature of the system as a rule-following machine. This distinction matters because it shapes what these systems can and

cannot be trusted to do, a point the chapter returns to repeatedly in the sections on risk and governance below.

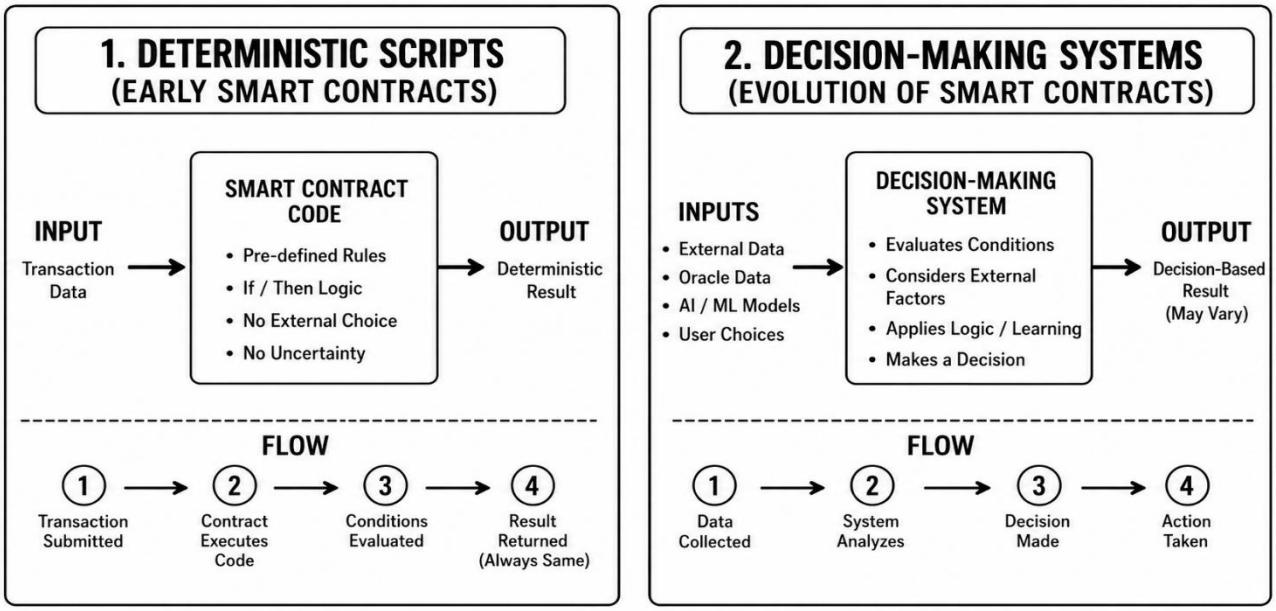


Fig 2: Deterministic Scripts to Decision Making Systems

10.3 ARCHITECTURE OF AN AUTONOMOUS DECISION FRAMEWORK

A typical autonomous decision framework layered on top of smart contract infrastructure has four components, and understanding each of them along with the reasons they are kept conceptually and often physically separate is essential to understanding both the power and the fragility of these systems.

10.3.1 The Data Ingestion Layer

Blockchains, by design, cannot natively access information that exists outside their own state. A blockchain node has no way to independently verify what the current price of oil is, whether a particular shipping container has arrived at a port, or what the temperature was at a given location yesterday. This is not a bug; it is a direct consequence of the same determinism discussed above: if nodes could each independently fetch external data, different nodes might fetch different values at different times, and the network would lose the shared, verifiable state that makes it useful in the first place. Oracles exist to solve this problem. An oracle is a service or, more robustly, a decentralized network of independent services that fetches external data and submits it on-chain in a form the contract can consume. Because any single oracle could be compromised, go offline, or simply be wrong, well-designed oracle networks aggregate and cross-validate inputs from multiple independent sources before delivering a value on-chain, and often include mechanisms for detecting and penalizing sources that report outlier or manipulated data. Some oracle networks also incorporate a delay or a dispute window, allowing a submitted value to be challenged before it becomes final, which trades a small amount of latency for a meaningful increase in reliability. The data ingestion layer is, in practice, the most common point of failure for autonomous decision frameworks, a point this chapter returns to

in Section 10.5 precisely because it is the layer where the system's deterministic, verifiable internal logic meets the messy, unverifiable external world.

10.3.2 The Decision Logic Layer

This is where evaluation happens: the layer that takes the data supplied by the ingestion layer, along with any relevant on-chain state, and produces a decision. Implementations range enormously in sophistication. At the simplest end are threshold rules: release funds if delivery is confirmed by three or five oracles; liquidate a position if collateral value falls below 110% of the loan value; pay out an insurance claim if a verified rainfall sensor records less than a specified number of millimeters over a specified period. These rules are easy to audit, easy to explain to the parties bound by them, and easy to reason about formally which is precisely why they remain the dominant approach in high-value, high-stakes applications. More elaborate systems combine multiple weighted signals into a composite score, then act based on where that score falls relative to one or more thresholds. A protocol assessing counterparty risk, for example, might combine a borrower's collateralization ratio, their transaction history, the liquidity of their posted collateral, and broader market volatility into a single risk score, adjusting its behavior the interest rate it charges, the collateral it demands, the speed with which it intervenes as that score changes. At the most experimental end, some systems incorporate off-chain machine learning models: a model trained to detect fraudulent insurance claims, forecast short-term price movements, or classify the likely cause of a supply chain disruption. Because on-chain computation is expensive and most machine learning models are far too large and complex to run economically inside a blockchain's execution environment, these systems typically compute the model's output off-chain and then submit only that output along with some form of proof or attestation that the computation was performed correctly and honestly back on-chain, where the contract acts on it as though it were any other oracle-supplied data point. This pattern, sometimes called an "AI oracle," is one of the fastest-moving and least standardized areas of the field, and it inherits both the promise of machine learning (the ability to detect patterns no explicit rule could capture) and its characteristic weaknesses (opacity, sensitivity to training data, and the difficulty of formally verifying a model's behavior in advance).

10.3.3 The Execution Layer

Once a decision is reached, the contract executes the corresponding state change: transferring assets, updating a registry, minting or burning tokens, triggering a downstream contract call, or emitting an event that off-chain systems can act on. This layer is, by design, the most mechanical and least interesting part of the architecture and that is exactly the point. By the time execution happens, all of the genuinely difficult reasoning has already taken place in the decision logic layer; execution is meant to be a faithful, uncontroversial translation of a decision into a state change, with no discretion left to exercise. Keeping execution mechanical and separate from decision logic also has a practical security benefit: it narrows the attack surface. If an attacker wants to manipulate a contract's outcome, they generally need to attack the decision logic layer or the data feeding into it the execution layer itself, having no judgment to corrupt, offers comparatively little to attack directly.

10.3.4 The Governance and Override Layer

Many systems maintain a human or multi-party checkpoint midway in the pipeline between decision and execution since fully autonomous execution carries actual risk—a point covered in detail in Section 10.5. This can take many different forms: a multi-signature veto, which requires a specific group of parties to actively block an action rather than actively approve it; a timelock, which delays execution by a predetermined amount of time during which the decision is publicly visible and can be challenged; or a decentralized governance vote, carried out by token holders or another defined constituency, which can pause, reverse, or override an autonomous decision before or shortly after it takes effect. The conflict between safety and autonomy is most apparent at the governance layer. Each checkpoint that is added to this layer slows down the system and increases its reliance on the integrity and diligence of the person with override power, but it also offers a chance to identify and fix a mistake, a manipulated input, or an unexpected edge case before it has an irreversible effect. The range of governance structures based on this tradeoff is covered in detail in Section 10.6.

10.3.5 Why the Layers Are Kept Separate

Even while the particular implementation details differ greatly, this intentional four-layer separation of ingestion, decision, execution, and governance is present in almost every significant autonomous decision framework in production today. Because the interface between the two layers is well-defined and can be reasoned about independently, a security researcher studying a protocol's decision logic does not also need to concurrently verify the correctness of every oracle it depends on. Additionally, it allows for the replacement or upgrade of a single layer without rewriting the entire system. For example, it can be used to tighten the governance layer's veto window in response to a near-miss without changing how decisions are made, or it can be used to swap in a more reliable Oracle network without affecting the decision logic that uses its data. After ten years of experimenting, most of it through the analysis of costly failures, this modularity is, in many ways, the most significant architectural lesson the discipline has learnt.

10.4 CATEGORIES OF DECISION FRAMEWORKS

Today's decision frameworks can be broadly divided into four overlapping groups based more on the location of the system's "intelligence" than on the underlying technology.

10.4.1 Rule-Based Frameworks

Rule-based frameworks, which are still by far the most popular method in high-value applications, embed clear if-then logic that has been agreed upon beforehand by all parties. A rule-based framework is a supply-chain payment contract that releases payments when a shipment's GPS coordinates match a delivery location and a correlating sensor verifies an undamaged temperature-controlled seal. This also applies to bonds that automatically pay out when a referenced interest rate surpasses a predetermined threshold on a given date. Transparency is the main advantage of rule-based frameworks: each participant can read the rules beforehand, comprehend precisely what will cause what result, and confirm that the

contract performed as intended after the fact. Because the rules are set at deployment, no party can subsequently argue that the contract operated in a way they did not agree to, unless there was an obvious bug. This transparency is also, in a sense, a sort of consent. The main drawback, which is covered in detail in Section 10.2, is rigidity: all relevant scenarios must be explicitly anticipated and encoded, and any scenario that deviates from the encoded rules either has no defined behavior or defaults to a behavior that was not intended for that particular situation.

10.4.2 Threshold and Consensus Frameworks

Oracle reports, validator votes, and sensor readings from geographically dispersed devices are all aggregated by threshold and consensus frameworks, which only take action when a predetermined quorum or confidence level is reached. This is essentially a generalization of the rule-based approach that forgoes some simplicity in favor of significantly higher robustness: the contract requires agreement, or near-agreement, among multiple independent sources before acting, as opposed to relying solely on one data source. This category works especially effectively in scenarios when obtaining a small number of extra confirming signals is inexpensive and there is a high danger of a single corrupted or incorrect input. In order to significantly lower the likelihood that a single malfunctioning sensor or a single compromised data provider will result in an inaccurate payout, a weather-based insurance product might, for instance, require verifying rainfall data from three independently managed weather stations rather than just one. Threshold frameworks are often saved for judgments where the value at stake warrants the additional overhead because gathering and reconciling several signals requires more time and money than relying on just one.

10.4.3 Market-Driven Frameworks

Market-driven frameworks allow choices to be made based on the design of incentives rather than on explicit guidelines for what constitutes the "correct" result. The most obvious example is automated market makers, which determine asset prices algorithmically based on the proportion of assets held in a liquidity pool; no rule in the system specifies what the "correct" price of an item is at any particular time. Rather, price is a result of the autonomous, self-serving behaviors of several participants, including traders, arbitrageurs, and liquidity providers, all of whom react to the incentives the system generates. In a crucial way, this is the least "decision-making" of the four categories since no single system component is assessing circumstances and selecting options in the same manner as a decision logic layer in the other categories. However, because a well-designed market-driven framework does not rely on any one party's assessment of the situation being accurate, its overall behavior can be remarkably responsive to real-world conditions, frequently more responsive, and more resistant to manipulation by any single actor than a rule-based or threshold framework would be. The trade-off is that market-driven frameworks are only as trustworthy as the market they generate; if the incentives are poorly crafted or the market is so thin that a single, well-resourced participant can influence it, the ensuing "decisions" may be manipulated or exhibit behaviors that deviate significantly from the system's designers' intentions.

10.4.4 Model-Assisted Frameworks

The newest and least standardized of the four types are model-assisted frameworks, which use predictions or classifications from off-chain machine learning models as described in Section 10.3.2. Examples of this category in active use or development include a forecasting model that modifies a protocol's risk parameters ahead of expected market stress, a credit-scoring model that estimates a borrower's likelihood of default from a wider set of signals than a simple collateralization ratio can capture, and a fraud-detection model that flags suspicious insurance claims for additional review. The ability of model-assisted frameworks to identify patterns that no human drafter would consider encoding as an explicit rule and to do so at a level of nuance that threshold and consensus techniques find difficult to reproduce is what makes them appealing. The corresponding risk, which is covered in more detail in Section 10.5, is that this pattern-detection capability comes at the expense of explainability: a model's output is typically not reducible to a straightforward, auditable rule, making it significantly more difficult for an impacted party, a court, or a regulator to ascertain after the fact whether a particular decision was reasonable or whether it reflected some bias or flaw ingrained in the model's training. Today, model-assisted frameworks are almost always deployed with a governance checkpoint discussed in Section 10.6 rather than being given unsupervised authority over consequential outcomes because it is still a genuinely unsolved problem at scale to verify that a submitted model output is trustworthy without simply trusting whichever party ran the computation.

10.4.5 Choosing Among Categories

In practice, most production systems do not fit neatly into a single category; they combine elements of several. A lending protocol might use threshold-based oracle consensus for its core price feed, rule-based logic for its liquidation trigger, and a market-driven interest rate curve that responds to pool utilization, all within the same system. The choice of which category to lean on for which decision is, in essence, a judgment about where the greatest source of risk lies: rule-based logic where predictability and auditability matter most, threshold consensus where a single point of failure would be unacceptable, market mechanisms where no central authority should be trusted to set an outcome, and model-assisted approaches where the relevant pattern is too complex to reduce to an explicit rule accepting, in that last case, a corresponding reduction in transparency.

10.5 RISK, TRUST, AND FAILURE MODES

Understanding these failure modes is crucial to comprehending why the sector has generally shifted toward tiered governance rather than complete autonomy for high-value decisions. Autonomy brings new failure modes that fully deterministic, narrowly scoped contracts do not have.

10.5.1 Oracle Manipulation

Even while the contract code itself performs exactly as stated and would pass any audit of its internal logic with flying colors, if a decision relies on other data, corrupting or manipulating that data source could corrupt the conclusion. This is, in a way, the sneakiest kind of failure since it doesn't even need discovering a flaw in the contract; all it takes is figuring out how to provide it false information that it can't independently verify. Historically, attacks of this type have taken

many forms, such as taking advantage of a short window between the time a price is published and when a contract acts on it, compromising a single oracle operator, or manipulating a lightly traded market to momentarily skew a price feed. Although they come with some additional latency and cost, the threshold and consensus frameworks covered in Section 10.4.2 are expressly designed to mitigate this type of risk.

10.5.2 Model Opacity

As covered in Section 10.4.4, when a decision is based on the output of a machine learning model, even the individuals who developed and implemented the model may not be able to completely explain the logic behind a particular decision, let alone the party impacted by its output. This makes real-time auditing more challenging since it is hard to confirm ahead of time that a model will act rationally in all possible scenarios. It also makes it more difficult to resolve disputes after the fact because a party contesting a decision lacks a straightforward rule to point to as having been applied incorrectly. Naturally, model opacity is a well-known issue in the larger field of applied machine learning and is not exclusive to blockchain-based solutions. The autonomous-contract context is unique in that a model's output, once acted upon by a smart contract, may cause an irreversible on-chain consequence much more quickly and with much less chance for human review than the same model's output would in a traditional software system with a human reviewer in the loop.

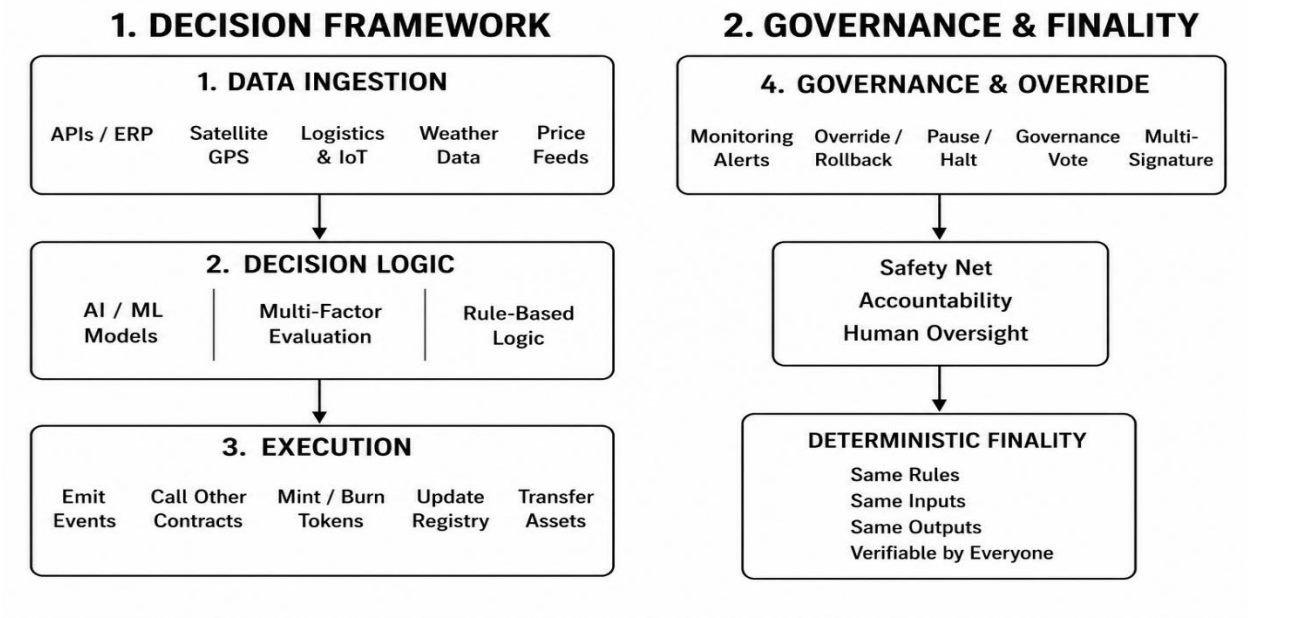


Fig 2: Architecture of an Autonomous Decision Framework

10.5.3 Cascading Automation

An error or a manipulated input in one system can spread quickly through an entire network of dependent contracts, faster than any human reviewer could realistically intervene, when autonomous contracts trigger other autonomous contracts. This is a common pattern in interconnected decentralized finance systems, where one protocol's output frequently serves

as another protocol's input. For example, a price feed manipulation that causes an improper liquidation in one protocol may feed straight into another that interprets the liquidation as a legitimate signal, exacerbating the initial issue before anyone has a chance to detect or react. One of the best arguments in favor of the governance checkpoints covered in Section 10.6 is that a delay of even a few minutes, imposed at a single well-chosen point in a chain of dependent contracts, can be sufficient to prevent a localized error from becoming a systemic one. This dynamic is somewhat similar to the cascading failures that have occasionally struck highly automated segments of traditional financial markets.

10.5.4 Irreversibility

Blockchain transactions are typically final: once a transaction is confirmed and buried under enough subsequent blocks, reversing it is, for practical purposes, impossible without the extraordinary and highly disruptive step of coordinating a fork of the underlying network an option that is essentially never available to an individual protocol and is reserved, if it is available at all, for the most catastrophic, network-wide emergencies. A decision framework that acts autonomously on flawed input may therefore execute an outcome that cannot be undone, in sharp contrast to a human-mediated process, where a mistake identified shortly after the fact can typically still be corrected, reversed, or renegotiated before it causes lasting harm. This asymmetry the ordinary reversibility of human decision-making against the ordinary finality of on-chain execution is arguably the single most important reason that thoughtful system designers hesitate to grant autonomous frameworks unsupervised authority over genuinely high-stakes, high-value decisions, even when the decision logic itself has been extensively tested and appears sound.

10.5.5 Correlated and Systemic Risk

A less obvious but increasingly important failure mode arises when many independent protocols come to rely on the same small set of underlying components: the same oracle network, the same handful of widely used model providers, the same category of collateral asset. In this situation, a single point of failure is no longer contained within one system; it becomes a source of correlated risk across many superficially unrelated protocols simultaneously. This is analogous to the systemic risk that regulators in traditional finance worry about when many institutions share exposure to the same underlying assets or counterparties, and it is a growing area of concern as the decentralized finance ecosystem has matured and consolidated around a smaller number of dominant oracle and infrastructure providers than existed in its earlier, more fragmented years.

10.6 GOVERNANCE MODELS FOR AUTONOMOUS SYSTEMS

Because full autonomy carries real risk, most deployed systems adopt a spectrum of governance rather than an all-or-nothing approach, and the choice of where to sit on that spectrum is, in the end, always a bet about which is more costly for a given decision: the friction of a checkpoint, or the risk of an uncorrected error.

10.6.1 Human-in-the-Loop

In a human-in-the-loop design, the contract proposes an action, but a human or a multi-signature group must actively approve it before execution proceeds. This is the most conservative model on the spectrum, and it is generally reserved for decisions that are both high in value and low in frequency, a protocol's treasury allocation, for instance, or a major parameter change that will affect every user of the system. The obvious cost is speed: a human-in-the-loop system cannot respond to a rapidly developing situation any faster than its designated approvers can be reached and can review the proposed action, which may be entirely unsuitable for time-sensitive decisions like liquidations in a fast-moving market.

10.6.2 Human-on-the-Loop

In a human-on-the-loop design, the contract executes automatically, but within a monitoring window during which a designated party or, more robustly, any member of a defined community can veto or pause the action before it becomes final. This model attempts to capture much of the speed advantage of full autonomy while retaining a meaningful safety net: the default behavior is action, not inaction, but that default can still be overridden if something looks wrong. Timelocks, which delay the execution of an approved but not-yet-finalized action by a fixed period, are the most common implementation of this pattern, and they have become close to a standard feature in serious decentralized protocols precisely because they offer a reasonable middle ground between responsiveness and safety.

10.6.3 Fully Autonomous

In a fully autonomous design, the contract executes without any external checkpoint at all. This model is typically reserved for decisions that are low in individual value, extremely well understood, and high in frequency, such that the cost of a governance checkpoint in delay, in the attention it demands from human reviewers, in the sheer number of times it would need to be exercised clearly outweighs the risk that any single instance of the decision is wrong. Routine, small-scale liquidations within a well-tested and heavily audited lending protocol are a common example: individually, each liquidation is small enough that an error is tolerable and recoverable at the level of the system as a whole, even though no individual liquidation is reviewed by a human before it happens.

10.6.4 Choosing a Governance Model

Choosing a governance model is, in the end, fundamentally a risk-management decision rather than a purely technical one, and it typically depends on three factors working together: the value at stake in a given decision, the frequency with which that decision needs to be made, and the reversibility of the consequences if the decision turns out to be wrong. High value, low frequency, low reversibility points strongly toward human-in-the-loop. Low value, high frequency, high reversibility (in the sense that any single error is small and easily absorbed by the system as a whole) points strongly toward full autonomy. Most real systems occupy some middle ground, and many of the more sophisticated ones vary their governance model dynamically, tightening oversight during periods of unusual market volatility or unusual on-chain activity, and relaxing it during calmer periods, in an attempt to get the benefits of both ends of the spectrum depending on prevailing conditions.

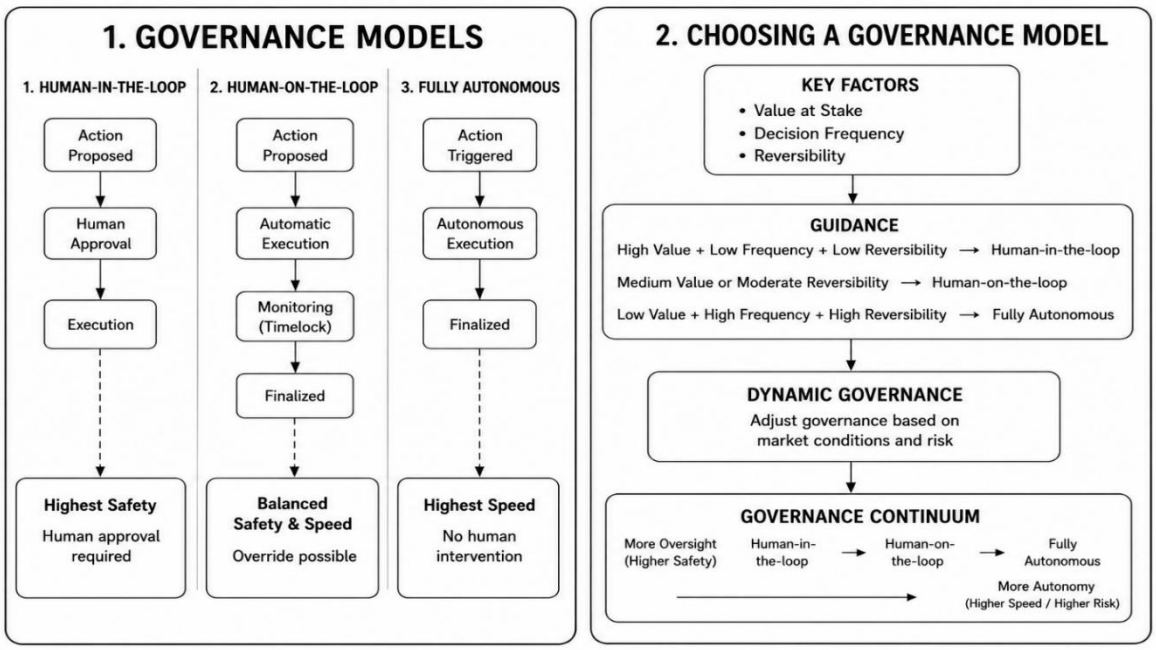


Fig 3: Governance model for Autonomous System

10.7 APPLICATIONS IN PRACTICE

The abstract architecture described above becomes clearer when grounded in the domains where autonomous decision frameworks have actually been deployed.

10.7.1 Decentralized Lending

Decentralized lending protocols were among the earliest and remain among the most mature applications of autonomous decision-making in the blockchain space. A lending protocol must continuously monitor the value of collateral posted against outstanding loans, decide when a position has become under-collateralized, and liquidate that position typically by allowing a third party to repay part of the loan in exchange for a discounted portion of the collateral quickly enough to prevent the protocol itself from accumulating bad debt. Every layer of the architecture described in Section 10.3 is visible here: an oracle network supplies price data; a decision logic layer compares that price against a collateralization threshold, often adjusted by a risk score that accounts for the collateral asset's volatility and liquidity; an execution layer carries out the liquidation; and, in more conservative protocols, a governance layer retains the ability to pause liquidations entirely during periods of extreme, possibly manipulated market volatility.

10.7.2 Parametric Insurance

Instead of requiring a claims adjuster to evaluate the particulars of each loss, parametric insurance products pay out automatically depending on a measurable, objective trigger, such as a rainfall level, wind speed, or flight delay. This is essentially a rule-based or threshold-based framework for making decisions in an area where human judgment has historically been delayed and subjective. Speed and predictability are appealing: a farmer whose crop insurance is activated by a validated drought index is paid in a matter of days instead of months, and they

don't have to demonstrate the severity of a specific loss. The corresponding limitation is the potential for the parametric trigger and the policyholder's actual loss to diverge, meaning that a farmer who actually suffered real damage might not receive anything because the particular rainfall threshold was not technically crossed, or they might receive a payout despite experiencing minimal actual harm. One of the main areas of current development in this application is reducing basis risk, typically through more numerous, localized, and finer-grained data sources feeding a threshold or consensus framework.

10.7.3 Supply Chain and Trade Finance

Physical-world sensor data, GPS position, temperature, humidity, tamper-evident seals, and financial triggers are all combined in supply chain systems, which automatically release payment when a cargo meets predetermined delivery and condition requirements. The tension between rule-based simplicity and real-world complexity is especially evident in these systems: a rule that states that payment releases upon arrival at a delivery location works well for the vast majority of shipments, but it struggles with the edge cases that matter most in practice, such as a shipment that is rerouted because of a port closure, a sensor that malfunctions mid-transit, or a partial delivery that is divided among several containers. Instead of trying to incorporate every possible logistics contingency into the contract itself, many production systems in this field have shifted toward hybrid designs: rule-based automation for the routine case, paired with a human-in-the-loop or human-on-the-loop escalation path for shipments that fail to satisfy the standard criteria cleanly.

10.7.4 Decentralized Autonomous Organizations

Decentralized autonomous organizations, or DAOs, use autonomous decision frameworks for organizational governance, such as grant funding, protocol parameter modifications, treasury management, and other decisions typically made by a company's board or management team, rather than for specific financial transactions. A human-in-the-loop or human-on-the-loop execution layer, such as a timelock that postpones the implementation of an approved proposal to give the community one last chance to object before it takes effect, is frequently combined with token-weighted voting, which is the primary decision mechanism used by the majority of DAOs. This framework can be either market-driven or consensus-based. Since governance-token voting is susceptible to a type of manipulation—the concentrated accumulation of voting power by a single well-resourced actor that closely resembles the oracle-manipulation risk facing more specifically financial applications—DAOs have also emerged as a significant testing ground for some of the risks covered in Section 10.5.

10.8 LEGAL AND REGULATORY CONSIDERATIONS

Existing contract law, which was created over centuries with a human decision-maker implicitly presumed at every point, was not intended to provide clear answers to the open legal concerns raised by autonomous decision frameworks. Liability is the most pressing of these issues. When an autonomous system makes a decision that results in harm, who is responsible—the developer who wrote the original code, the deployer who put it into production, the oracle operator who provided the data the decision depended on, the token holders who controlled a

parameter change that made the harmful outcome possible, or, in some cases, no one at all—based on the theory that the impacted party voluntarily interacted with an openly published, deterministic system and accepted its stated rules? Parties are left to rely on general principles of software liability, financial regulation, or consumer protection law that were not specifically written with autonomous decision-making in mind because different jurisdictions have approached this question differently and, in many cases, have not yet addressed it directly at all. The nature of the agreement itself is the subject of a second, related query. A certain level of predictability and mutual knowledge between the parties regarding what has been agreed upon are typically required by contract law. Does a contract whose terms automatically change over time in response to circumstances that were not known at the time of formation still constitute the kind of fixed, mutually understood agreement that contract law traditionally requires, or does this flexibility—no matter how explicitly stated at the outset—undermine the predictability that the law typically expects an enforceable contract to have? For model-assisted frameworks, where even the system's designers might not be able to fully foresee in advance how the model would act in a certain future event, this question becomes even more pertinent.

Regulatory classification is a third question that is becoming more and more important when autonomous frameworks are applied to financial products. Even though they are organized and function fundamentally differently, a parametric insurance product, a lending protocol, or a governance token may economically resemble a standard regulated financial instrument. Many jurisdictions' regulators have started to apply current financial regulatory frameworks to these products, but they also recognize that these frameworks, which are based on centralized institutions and human intermediaries, fit awkwardly onto systems that lack a clear central operator to regulate. The legal and regulatory environment supporting autonomous decision frameworks is still developing, frequently unevenly among jurisdictions and frequently well behind the rate of technological advancement, and none of these issues are nearly resolved. Instead, seeing this as a solved issue with a universally applicable solution, anyone implementing autonomous decision systems for significant, real-world agreements should treat it as a really unresolved topic requiring skilled legal assistance familiar with the particular jurisdiction.

CHAPTER 11

DECENTRALIZED AUTONOMOUS ORGANIZATIONS AND GOVERNANCE MODELS

Contributed by: Khakon Das, Sagar Das, Anirban Sinha, Ankita Mitra

11.1. INTRODUCTION

A Decentralized Autonomous Organization (DAO) is a blockchain-based system that enables people to coordinate and govern themselves mediated by a set of self-executing rules deployed on a public blockchain, and whose governance is decentralised (i.e., independent from central control). The studies that were looked at show that DAO and blockchain governance are changing in five stages. The basic idea behind DAOs is that they are decentralized ways of doing things that make everything more open and honest but they have problems with being truly decentralized working for a lot of people being secure and gaining trust. When look at all the research, it was observed that there are stages that governance goes through and that different people have different roles. We also see that we need to have rules that are fair. Some studies suggest that we should use ideas like shared resources, governing networks and mixing things that happen on and, off blockchain. We need to test these ideas in real life. When we look at what happens we see that voting systems where everyone has an equal say and where people can put their money towards things they care about make DAOs more sustainable. When a few people have too much power it hurts the growth of the DAO. New studies are pointing to DAOs that use intelligence and mix different things as the way of the future but we need to have better laws be able to handle more people and be more efficient for this to work. DAOs and blockchain governance are still. We need to keep working on DAOs to make them better. This report looks at the research that has been done on DAO and blockchain governance. It has gotten better over time. Can be broken down into five main parts that are all connected. The first part is about the ideas behind DAOs and how they can be seen as real organizations. The next part is about looking at all the information that's available and figuring out what is missing. Then there are proposals for ways to govern DAOs. After that there are studies that use data from DAOs to see how well these new ideas work. Finally, there are studies that think about the future and how DAOs can work with intelligence. These studies show us where the field is going. They have not been tested very much yet. Overall, the research on DAO and blockchain governance is moving forward in a way. We are going from thinking about the ideas to actually testing them and coming up with new things. It is important to keep doing

this research so we can really understand how DAO and blockchain governance work and make them better. DAO and blockchain governance research are getting better. That is good, for the future of blockchain systems.

A Decentralized Autonomous Organization is a system that uses blockchain to help people work together and make decisions on their own. This system has rules that are automatically carried out when certain conditions are met. These rules are on a blockchain so everyone can see them. This way people can trust each other. The system is safe. A Decentralized Autonomous Organization is like a community that runs itself without an authority in charge. A Decentralized Autonomous Organization is very transparent which means everything is out in the open. Decentralized Autonomous Organizations have become very important in the Web3 ecosystem. They help people make decisions together manage money in a way and let the community make decisions. Ethereum explained what a Decentralized Autonomous Organization is back, in 2015. Further the DAOs have gained popularity across decentralised finance (DeFi), gaming, investment funds, NFTs, and social communities. Furthermore, the basic concepts have been proposed by the Dan Larimer in the year 2013 and further it has been invented by the Vitalik Buterin in 2014. The proposed work has been done using open-source code by Christoph Jentzsch in 2016. DAO's operational rules are implemented through blockchain based smart contracts rather than traditional management structures. Unlike conventional organisations, where executives or boards make decisions, DAOs empower community members to participate directly in governance through voting mechanisms.

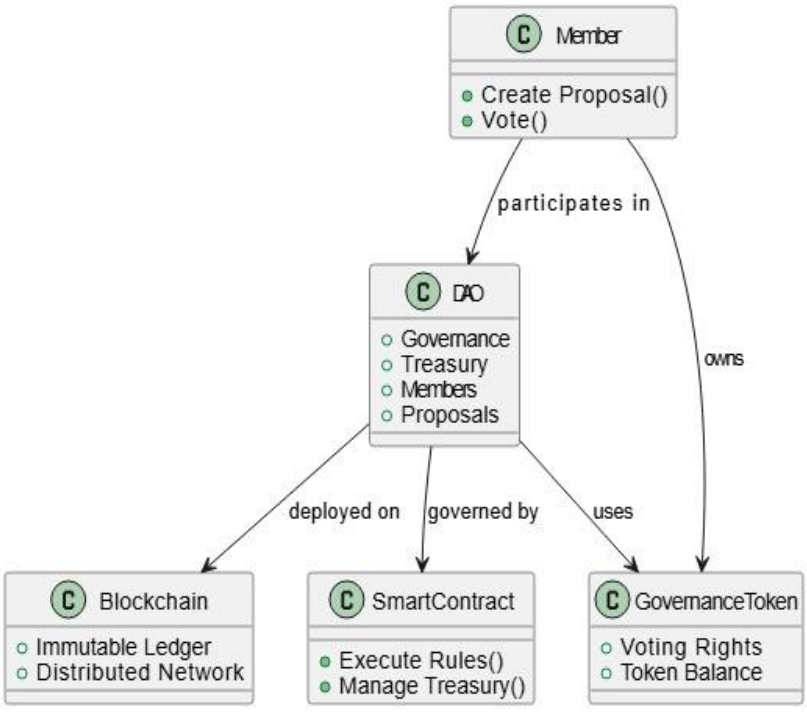


Figure 1: Parameters of the Decentralized Autonomous Organizations (DAOs)

This decentralized structure ensures transparency, fairness, and collective ownership. Core components property of Decentralized Autonomous Organization (DAO) described in below section. Here in figure 1 and figure 2, the authors' tries to depict the important parameters and the timeline of the evolution related to DAO respectively.

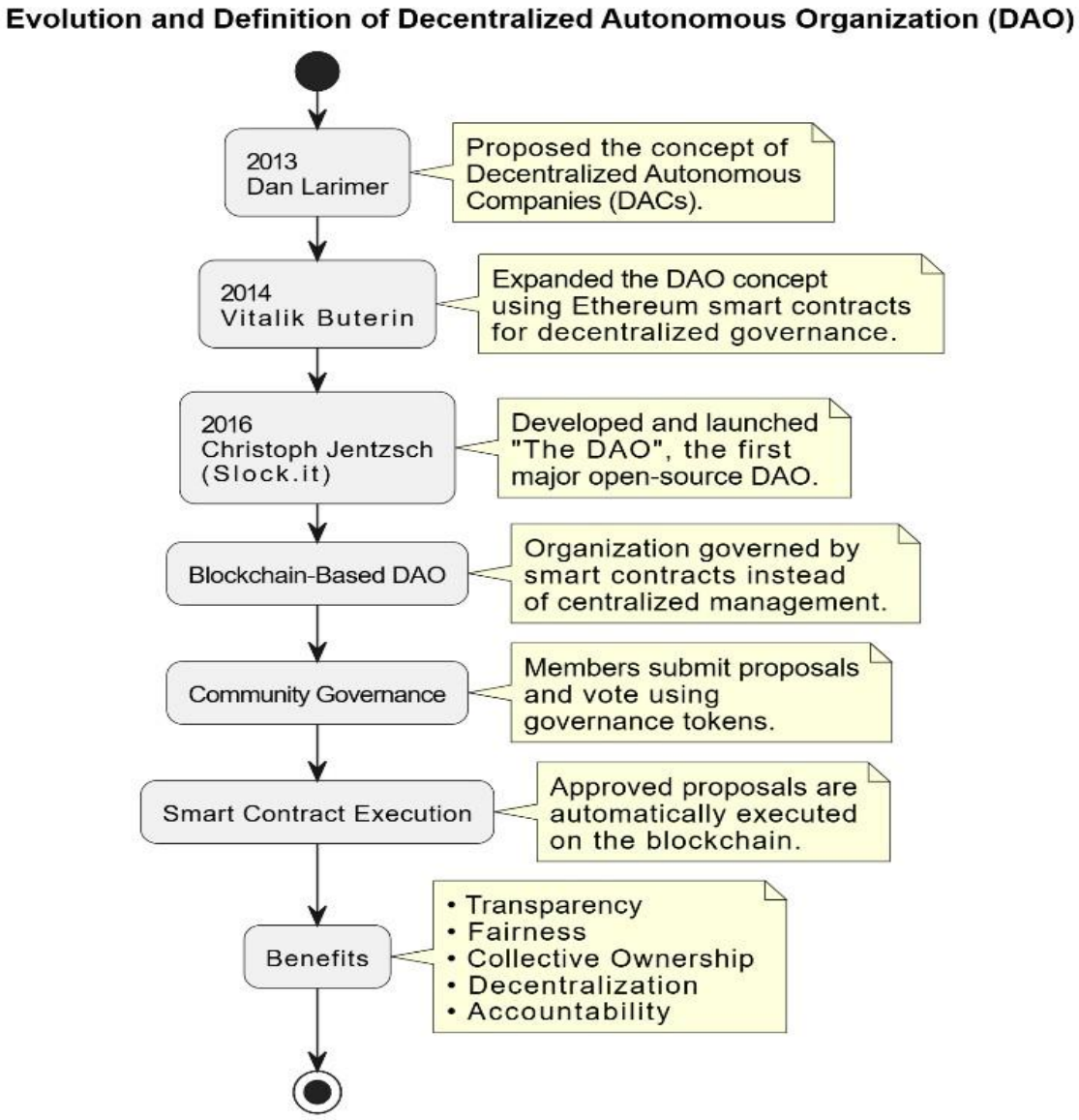


Figure 2: Timeline of the Evolution of Decentralized Autonomous Organizations (DAOs)

Core Components*:* The DAO has key parts that make it work in a decentralised way. It uses blockchain technology to store records and financial transactions. This makes everything transparent and unchangeable. The blockchain is like a ledger. Smart contracts are also crucial. They automate the rules of the organisation. Carry out decisions that have been approved. The best part is that they do this without needing any middlemen. Governance tokens are another component. They give members the right to vote on decisions. The DAO treasury is like a safe. It manages the community's funds using contracts. The community is made up of people. These include developers, investors, validators and contributors. They all play a role in suggesting and voting on decisions, about how the DAO should be run. The DAO relies on its community to make it work. Community members vote on governance decisions. The DAO uses governance tokens for voting. Further the functioning of a DAO begins with the deployment of smart contracts that define governance rules. Governance tokens are then distributed among participants, allowing them to submit proposals and vote on organizational decisions. Once voting concludes, smart contracts automatically verify the

results and execute approved proposals. This automated process eliminates the need for centralized authorities and significantly enhances transparency, efficiency, and trust within the organization. The DAOs' decision making and governance workflow has been shown in the figure 3.

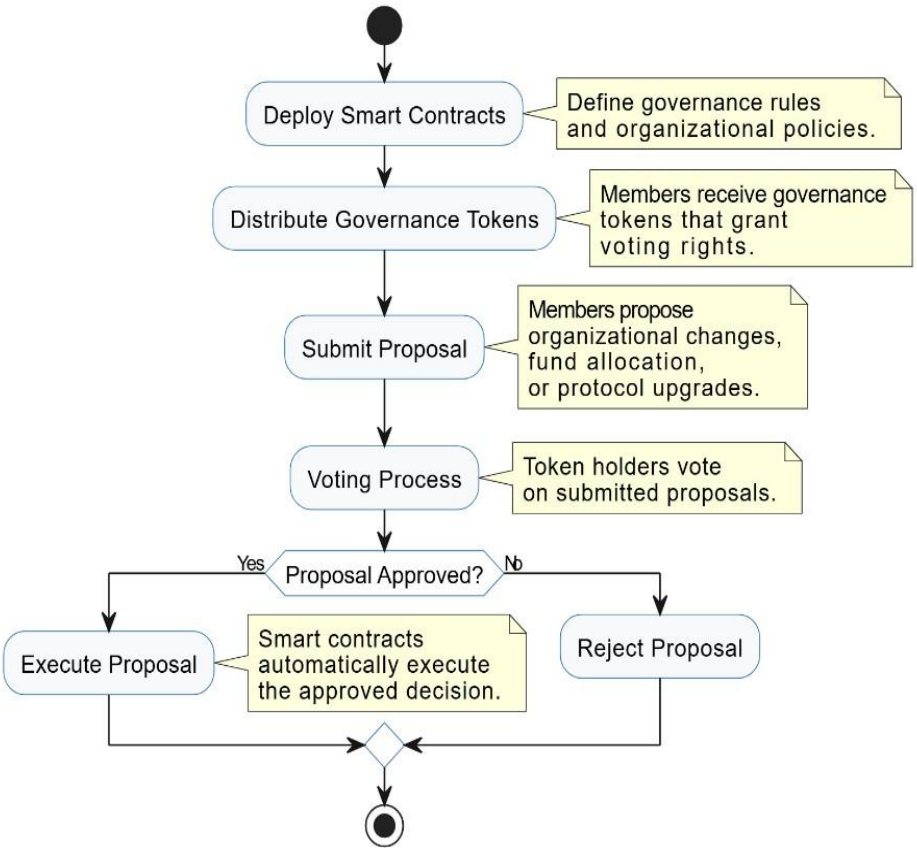


Figure 3: DAO Decision Making and Governance Workflow

Governance models decide how things get done in a DAO. They figure out how ideas are suggested voted on and put into action. The common way to do this is, with token-based governance. This means that people who own tokens have more say in what happens. There are ways to make decisions too. Some DAOs use a system where every person gets one vote. Others use delegated governance, where people pick someone to make decisions for them. Then there is voting, which is a bit more complicated. Some DAOs use reputation-based governance, where people who are well known and respected have more influence. Each of these methods tries to be fair get people involved and make decisions quickly. They also try to stop people from manipulating the system. Lately a lot of DAO projects are using a mix of these methods. They are combining approaches to make better decisions. This is called a governance model. The idea is to take the parts of each system and put them together to make something that works really well for the DAO. The applications of Decentralized Autonomous Organizations or DAOs are found over the place in many different areas. DAOs are used in the area of finance which is also known as DeFi to oversee things like lending protocols and exchanges that are not controlled by a central authority and also to manage the treasury. Investment DAOs are a way for people to put their money together to invest in assets and new companies. Communities that deal with Non-Fungible Tokens or NFTs use DAOs to take care of the property that belongs to the community and to work on projects

that the community is interested in. DAOs are also used in the gaming world to let players have a say in how the game's developed and how rewards are given out. Then there are DAOs that make it possible, for people who create things teach and are part of online groups to work together in a way that is not controlled by a central authority. DAOs are used in all these areas to help people work together and make decisions. The advantages of application tutorials are discussed below.

The advantages of Decentralized Autonomous Organizations or DAOs are really good. They are better than organizations in many ways. For one thing DAOs are very open and honest. Every decision that is made is written down on a blockchain for everyone to see. This means that DAOs do not have to answer to one person in charge. They also use contracts to get things done which helps cut down on costs. DAOs are good for people around the world. Anyone who has governance tokens can help make decisions for the organization no matter where they live. This helps build trust and makes sure that everyone is held accountable. It also gives the community a sense of ownership. The characteristics of DAOs, such as transparency and decentralization are what make them special. DAOs are, about working together and making decisions as a group, which is why they are so effective. In behave of the study; some challenges are faced that are covered in the next part.

Challenges that DAOs face-Low participation from voters makes governance less effective. When a few people hold tokens, they can control decisions. Bugs in contracts can cause financial problems. There is still uncertainty, about the laws that apply to DAOs in places. DAOs also struggle with Governance attacks, Scalability issues, Slow decision-making. These problems affect how well DAOs work in industries. DAOs have a lot of potential. These challenges hold them back. Governance, token distribution and smart contract security are areas to improve. DAOs need to address these issues to be more effective.

11.2. PAST STUDIES

In this section we have done a study covering the last ten years. In 2018 Hsieh et al. Did a study on an organizational model that uses blockchain technology. This model can change how organizations are run make decisions and work together using automated systems. It offers transparency, trust and efficiency. It also has problems like decentralized governance, mining concentration, interoperability and network monopolization. To make Decentralized Autonomous Organizations (DAOs) work better we need to:

1. Develop consensus mechanisms that can handle users.

2. Improve governance on the blockchain.

3. Make it easier for different blockchain networks to work together.

4. Create incentive models that use cryptography and economics.

We also need to do research that combines management, economics, cryptography and organizational science to make sure DAO models work and can be used with traditional organizations. The DAO model needs to be validated through interdisciplinary research to support its sustainable integration alongside traditional organizational structures and

blockchain technology. The blockchain technology should be studied in the context of DAOs. DAOs have the potential to redefine governance, decision-making and collaboration, through decentralized and automated mechanisms. The study of DAOs and blockchain technology is essential. Lumineau and other researchers did something in 2020. They found a way to get people to work together. This way helps people cooperate and coordinate with each other. It is different, from the ways that use contracts and relationships to govern people. Blockchain governance is special because it does not rely on one person or group to make decisions. Instead, it uses a system that is spread out and allows people to make decisions together. Lumineau and the other researchers started something. They showed that blockchain governance can change the way people work together in all kinds of industries. They looked at blockchain governance from the perspective of how organizations work. They think it has the potential to really change the way people collaborate. Blockchain governance is a way of doing things and Lumineau and the other researchers think it is worth looking into. They think blockchain governance can help people build trust and make decisions together. The study will look at how blockchain governance models work in organizations. It will compare these models to ways of governing and see which one is better. The study will also look at how blockchain governance models affect how well an organization does, how much people trust it and how it makes decisions. We need to do research to see if the ideas we have are really true. We also need to come up with ways to make blockchain governance work better in life. This means we have to figure out how to make it bigger how to make people accountable and how to get all the people involved to work. This will help more people use blockchain governance and make it work better in organizations that are not controlled by one person. Blockchain governance is what we are talking about. Blockchain governance is important. Blockchain governance will help organizations that use blockchain.

In the year 2020 Morrison and other people did a study about DAOs. This study showed that DAOs are a deal in the history of blockchain governance. It also showed that organizations that do not need trust and are controlled by contracts can work. The DAO had some problems with how it was governed and kept safe but it also made us think about how we govern things in a different way. We need to think about ways that organizations without a central leader can work together and trust each other. The study was, about DAOs. How they are changing the way we think about blockchain governance and DAOs. To make things better in the future people should do research on how to create stronger rules for organizations. They should try to combine rules with flexible systems that can adapt to changes. They should also work on making these systems more secure and come up with plans for emergencies. It is an idea to test out these new systems, like decentralized organizations by doing case studies and looking at what the law says. Decentralized organizations will be better because they will be stronger, safer and more sustainable. [9]. The Zwitter and Hazenberg did some research in 2020. They came up with an idea for governing decentralized networks. This idea is called network governance or Mode 3 for short. The Zwitter and Hazenberg thought of this because the old ways of governing were not working well for blockchain and digital ecosystems. The new framework takes into account the roles that people play how power works in different situations and how things work both on and off the blockchain. This makes it a better fit for networks and things like

Web3 applications. The Zwitter and Hazenbergs decentralized network governance is really, about making sure that blockchain and digital ecosystems work better. For further improvement, future research should empirically validate the proposed framework, investigate the interaction between on chain and off chain governance, and develop practical policy and governance tools that support dynamic stakeholder roles, digital regulation, and effective decision making. These improvements will strengthen governance models for block chain ecosystems and other decentralized digital platforms. In another study conducted by the Faqir Rhazoui et al. in 2021, they provide a comparative analysis of the three leading DAO platforms namely Aragon, DAOstack, and DAOhaus. In this study the authors evaluating their growth, user activity, voting behaviour, and fund management. The findings reveal significant differences in platform performance and highlight the evolving nature of DAO ecosystems. However, the results are limited by the use of aggregated statistics, the diversity of DAO communities, and the influence of early adopters, indicating the need for more comprehensive analysis. Further the study conducts detailed qualitative and quantitative investigations into individual DAO communities, voting mechanisms, stakeholder participation, and the influence of factors such as reputation and financial incentives. Additionally, research should explore platform-specific trends, governance effectiveness, and the impact of block chain infrastructure on organizational behaviour. These improvements will contribute to the development of more scalable, transparent, and effective decentralized governance models for future DAO ecosystems.

The studies I read show that new Web3 technologies are changing the world, both online and offline. This makes Decentralized Autonomous Organizations or DAOs a part of how organizations will work in the future. Even though DAOs are being used more and more in industries we do not really understand how they work. There is a difference between how people are using DAOs and what researchers are studying. We need to learn more about how DAOsre set up how they are governed and how they make decisions. To fix this problem researchers are working on a plan, for DAOs. They are making models looking at technologies and trying to understand how DAOs can make decisions on their own. All of this work is meant to help us understand DAOs better and make it easier to study them and use them in the world. Overall these studies are helping us learn more about DAOs, which will make it easier to use them safely and efficiently in the Web3 world. Web3 technologies and DAOs are really important so we need to keep learning about them. In the next section a special categorization of existing studies has been proposed.

11.3. WORKING PROCESS

In this section a special categorization of existing studies has been proposed based on various feature. Here specifically the five categories (groups) are introduced.

11.3.1 Group 1: Systematic Literature Reviews

A study by Liu et al. In 2022 looked at blockchain governance. The study is a review of many research papers. It says governance is key to making blockchain systems sustainable adaptable and successful in the run. The researchers looked at 37 studies. They found that

most blockchain governance is about planning, running and upgrading platforms. This is done through things like proposals for improvement voting, by stakeholders and changing protocols. The study also found that consensus protocols and incentive mechanisms are crucial. They help make on-chain governance work by getting stakeholders involved and making decisions. Blockchain governance relies on these to coordinate participation and decision making. Governance is essential for blockchain systems to succeed. It helps ensure blockchain systems are sustainable and adaptable. Further it finds areas that need more improvement. Moreover, research creates a governance plan for different blockchain parts. This plan clearly states the roles of people involved who makes decisions who's responsible and what motivates them. It also deals with the law, regulations and moral issues that come with governance. The suggested future plan includes creating a blockchain governance framework. It also involves making a catalogue of governance design patterns. These would improve openness, effectiveness, ability to grow and uniformity, across blockchain systems. These changes are expected to make governance models stronger. They will help blockchain technologies become widely used and continue to develop. Blockchain technologies will become more reliable. Blockchain will be used often. It will improve over time. Further in the year 2022 another literature review has been performed by Santana et al., in this literature review the author try to enhances the understanding of Decentralized Autonomous Organizations (DAOs) by clarifying their core principles, organizational performance, and governance mechanisms while identifying key directions for future research. The study contributes by establishing a stronger theoretical foundation for DAOs, integrating multiple theoretical perspectives, and proposing an integrative model that explains the interactions between block chain technology, decentralized governance, human–machine collaboration, and institutional change. It also highlights the business opportunities, governance challenges, and ethical concerns associated with DAOs, emphasizing the need to prevent unethical and illegal activities within decentralized ecosystems. The thing about the research that was done before is that it does not cover everything. It mostly looks at what has been published in papers and that can be a problem. This means that we need to do research and look at things from different angles. We should do some real-life studies. Work with people from other fields to make sure our ideas are good. We also need to look at the nine areas that need research and see how they can really help people. We need to make sure we are doing the thing and being responsible. If we do all of this, we will have an understanding of DAOs and how they can be used in a good way. This will help DAOs be used in a way that's good for everyone, in the Web3 ecosystem. We want DAOs to be used in a way that's sustainable and fair.

11.3.2 Group 2: Foundational Theoretical Studies

Lumineau and others did a study in 2020. They found a way to help people work together. This way helps people cooperate and coordinate with each other. It is different from ways of working together. Old ways used contracts and relationships to make sure people worked together. Blockchain changes how organizations work together. It helps build trust. It helps make decisions as a group. This study helps us understand blockchain governance. It shows how blockchain can change how people work together in industries. Lumineau and others used science to study blockchain governance. Blockchain governance can transform how

people work together. It can change areas of business. Lumineaus study is important, for understanding blockchain. The researchers Zwitter and Hazenberg did some research in 2020. They came up with a way to govern networks in a decentralized way, which they call Mode 3. This new framework is meant to fix the problems with the ways of governing networks especially when it comes to blockchain and digital ecosystems. The Zwitter and Hazenberg framework takes into account the different roles that people play how power relationships change depending on the situation and how decisions made on and off the network affect each other. This makes the framework a better fit for networks that are decentralized (Mode3) like the ones used in Web3 applications and blockchain. The decentralized network governance framework proposed by Zwitter and Hazenberg is a deal, for blockchain based and digital ecosystems. The Faqir Rhazoui and his team did a study in 2021. They looked at three DAO platforms: Aragon, DAOstack and DAOhaus. The Faqir Rhazoui and his team wanted to see how these DAO platforms were doing. They checked things like how many people were using them what people were voting for and how they were managing their money. The study by the Faqir Rhazoui and his team found some differences in how well each DAO platform was working. This shows that DAO ecosystems are always changing. The Faqir Rhazoui and his team found this out by studying Aragon, DAOstack and DAOhaus. The Faqir Rhazoui and his team also found that their study was not perfect. They were looking at groups of numbers not individual DAO communities. The people who started using DAO platforms first had a big influence on the results. So, the Faqir Rhazoui and his team think we need to study DAO platforms, like Aragon, DAOstack and DAOhaus closely.

11.3.3 Group 3: Framework & Conceptual Model Proposals

Improvement Report Summary study by Van Vulpen and others was done in the year 2023. This study created a plan for open-source software producing DAOs. The plan has seven parts: the structure of roles and leadership in DAOs the way decisions are made in DAOs, the legal and ownership rules for DAOs the way members are managed in DAOs, the incentives for people in DAOs the management of the community in DAOs and how all these things work with the process of developing software in DAOs. This plan gives advice, for building and managing successful DAO communities. Further the DAO developers and researchers should refine governance designs, strengthen security against potential attack vectors, and validate the framework through additional real-world case studies. Continued research on governance effectiveness, legal structures, incentives, and community coordination will be critical to supporting the sustainable adoption of DAOs in future organizational, financial, and regulatory systems [16]. Li and Chen did some work in 2024 that helps us understand how DAO governance works. They came up with a framework for governance that is based on commons principles. This framework is supposed to make community engagement. It also helps with managing resources that everyone shares. It makes decision making a group effort. The goal is to have a system that is governed in a way. The framework tries to solve some problems. For example, it tries to keep people trusting each other. It also tries to keep the quality of the system high. It tries to make sure the system is working well. All of this is supposed to help Web3 ecosystems grow. Li and Chens framework is important for DAO governance. DAO governance is what this framework is,

about. In another study by Polcumpally et al., (2024), emphasizes that block chain governance shifts part of decision-making authority from humans to code, making trust a critical yet insufficiently explored aspect of decentralized governance. It highlights the need for stronger theoretical frameworks that integrate trust, governance, risk theory, and block chain to better explain the social and organizational implications of block chain-based systems [18]. For further improvement, future research should focus on developing trust centered governance models, validating frameworks through practical methods such as regulatory sandboxes and homeostatic interactions, and examining how delegated machine-based decision-making influences transparency, accountability, and long-term governance effectiveness.

11.3.4 Group 4: Empirical & Quantitative Studies

Rikken et al. in 2023 has been perform a study identifies key governance elements that influence the long-term viability of Decentralized Autonomous Organizations (DAOs). The findings show that democratic governance and non-weighted voting systems (one member–one vote) significantly improve DAO sustainability, while explicit financial or reputationbased incentives may be less effective than intrinsic motivation. These insights provide practical guidance for designing transparent, inclusive, and accountable DAOs [15]. In a work proposed by Bellavitis and Momtaz in the year 2025, the authors try to demonstrate that on-chain voting governance significantly improves DAO fundraising success by enhancing transparency, accountability, and stakeholder trust. In contrast, off-chain voting reduces investor confidence, particularly in larger DAO communities, where transparency and verifiability are critical for effective governance. In further improvement, DAO developers prioritize transparent on chain voting mechanisms, strengthen safeguards against voting collusion, and implement governance policies that promote openness and accountability. Future research should examine the long term performance of DAOs, evolving community dynamics, and the effects of transitioning between off-chain and onchain governance to develop more sustainable and efficient governance models. In another study done by Han et al. in 2024, improves the understanding of DAO governance by identifying how concentrated voting power among large token holders (whales) can negatively affect long-term platform growth. The findings show that governance risks can be reduced through larger platform adoption, token illiquidity, and staking-based voting mechanisms that encourage long-term participation and discourage opportunistic behaviour [19].

11.3.5 Group 5: Applied & Emerging Future-Oriented Studies

Chambefort and Chaudey in the year 2024 in their study, they highlight the rapid growth of Decentralized Autonomous Organizations (DAOs) as a transformative organizational model that enables decentralized governance, democratic decision-making, and innovative business models. It identifies DAOs as a promising evolution in organizational economics with the potential to reshape traditional corporate governance. Further in this research evaluate various strategies, voting mechanisms, governance efficiency, agency issues, and the comparative performance of DAOs versus traditional organizations. Developing stronger governance frameworks, improving decision making processes, and identifying

industry specific applications will enhance the effectiveness, scalability, and long-term sustainability of DAOs. In another study proposed by Pawlowski in 2024, highlights the significant potential of Decentralized Autonomous Organizations (DAOs) as scalable governance tools for managing complex systems, supporting sustainable development, and enabling decentralized collaboration. It suggests that DAOs will gradually integrate with traditional organizations through hybrid governance models, while advances in artificial intelligence (AI) will further enhance DAO efficiency, automation, and decision-making. Here further efforts focus on developing hybrid legal and governance frameworks, strengthening AI integration in DAO operations, and addressing challenges related to scalability, regulation, and decentralization. These improvements will support the secure, sustainable, and widespread adoption of DAOs in the evolving Web3 and digital economy. The chart below summarizes the studied researches recommendations made across all 16 reviewed papers, grouped into common themes, has been shown in the figure 4. Empirical validation through real world case studies is by far the most frequently requested direction, followed by stronger stakeholder accountability and clearer legal regulatory frameworks.

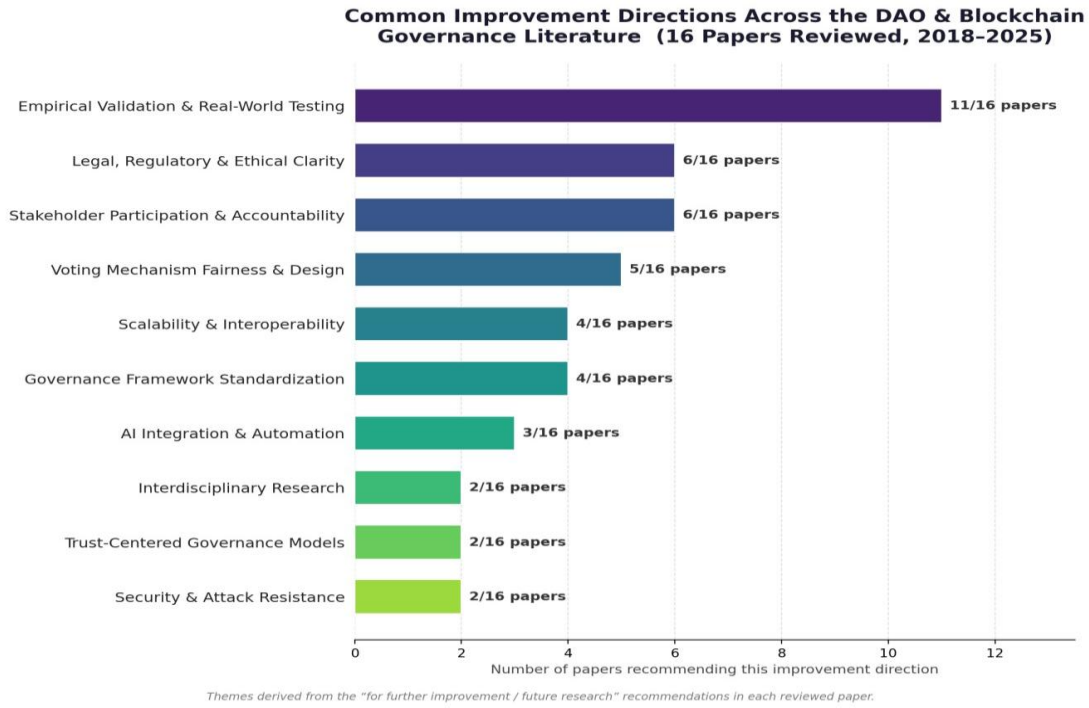


Figure 4: Summarization of research studies about DAO

11.4. STUDY'S OUTCOME AND ANALYSIS

This section organizes the studies below into five groups, based on the quality and type of research contribution each paper makes to the field of DAO and block chain governance. In the figure 5 a pictorial representation of the proposed categorisation has been shown.

Proposed categorisation:

Group 1: Foundational Theoretical Studies: Early, widely-cited works that established DAOs and blockchain governance as legitimate objects of organizational study. Highly conceptual studies influence though largely non empirical.

Group 2: Systematic Literature Reviews: Papers that synthesize a large body of prior research using explicit review methodology, offering the broadest and most methodologically rigorous overview of the field.

Group 3: Framework & Conceptual Model Proposals: Studies that introduce new governance architectures or theoretical models. Medium-to-high quality; contribution depends on later empirical validation.



Figure 5: Pictorial representation of the proposed groups

Group 4: Empirical & Quantitative Studies: The strongest evidentiary tier papers using real DAO platform data, voting records, or comparative statistics to test governance claims.

Group 5: Applied & Emerging / Future Oriented Studies: Recent, forward-looking papers (e.g., AI integrated governance, hybrid models) those are high in relevance but still light on empirical grounding.

11.5 CONCLUSION

This improvement report demonstrates that research on DAOs and block chain governance has matured into a structured, multi-tiered field with distinct yet interdependent contributions. The grouping of 16 studies across five quality tiers reveals both the intellectual foundations and current trajectory of the discipline. Foundational theoretical studies in Group 1 established DAOs as legitimate objects of organizational inquiry, providing the conceptual vocabulary and frameworks that subsequent work builds upon. While largely non-empirical, their high conceptual influence remains essential for situating new findings. Systematic literature reviews in Group 2 offer the broadest and most methodologically rigorous syntheses, consolidating fragmented research and identifying gaps for future investigation. Together, Groups 1 and 2 form the field's knowledge backbone. Groups 3 and 4 represent the shift from theory to design and validation. Framework and conceptual model proposals introduce novel governance architectures whose value depends on later empirical testing, while empirical and quantitative studies constitute the strongest evidentiary tier by using real DAO data and voting records to test governance claims. This tier is critical for moving the field beyond speculation toward evidence-based design. Finally, applied and future-oriented studies in Group 5 highlight emerging directions such as AI-integrated and hybrid governance models. Though currently light on empirical grounding, their relevance signals where the field is heading. Collectively, the five groups show a healthy progression: conceptual grounding, methodological synthesis, model building, empirical validation, and forward-looking innovation. Sustaining this pipeline will be key to advancing both theory and practice in block chain governance.

CHAPTER 12

TOKENIZED INCENTIVE MECHANISMS FOR INTELLIGENT NETWORKS

Contributed by: Krishnakash Giri, Sourasish Chakraborty, Priya Majumder, Disha Khan

12.1 INTRODUCTION

This chapter develops a rigorous architectural and game-theoretic framework for tokenized incentive design in autonomous intelligent networks — edge-computing swarms, federated-learning consortia, and decentralized IoT ecosystems. We formalize the mechanism-design problem underlying decentralized coordination, derive dominant-strategy-incentive-compatible reward structures for collaborative intelligence, and specify a complete Proof-of-Contribution protocol for tokenized decentralized federated learning, including Shapley-value-based data valuation, zero-knowledge contribution verification, and an algorithmic reward/slashing settlement procedure. We further provide an in-depth cryptoeconomic security analysis of collusion, bribery, and MEV-based attack vectors against tokenized intelligent networks, together with a mitigation matrix spanning token-curated registries, reputation-weighted governance, verifiable random functions, and hardware-enforced trusted execution. The chapter concludes by examining cross-chain swarm interoperability and AI-driven algorithmic tokenomics as emerging research horizons.

12.1.1 Architectural Paradigm & The Convergence of Intelligence and Decentralization

The last decade has witnessed the maturation of two technological trajectories that, until recently, evolved largely in isolation: the distributed autonomy of blockchain-based crypto-economic systems, and the proliferation of intelligent, semi-autonomous computational agents embedded in edge devices, sensor swarms, and federated machine-learning pipelines. We define an Intelligent Network as a topologically distributed system composed of heterogeneous, resource-constrained, and independently owned computational nodes—edge devices, IoT sensors, autonomous AI agents, and federated learning participants—that must coordinate to produce emergent collective intelligence (e.g., a shared predictive model, a swarm-level control policy, or an aggregated sensory picture of a physical environment) without recourse to a single trusted coordinating authority. This definition is deliberately broad because the incentive-design problem it poses is structurally identical across domains: a fleet of autonomous vehicles sharing occupancy-grid data, a federation of hospitals jointly training a diagnostic model, and a

mesh of environmental sensors reporting air-quality telemetry are all instances of the same underlying coordination problem. Each node possesses private information (its local data distribution, its true computational cost, its sensing accuracy) that is not directly observable by its peers or by any central planner. Each node is also a rational, self-interested economic actor—whether that self-interest is programmed directly into an autonomous agent's objective function or is the emergent behavior of a human-operated node seeking to minimize its operating costs. The historical default for coordinating such networks has been centralized orchestration: a cloud server aggregates sensor telemetry, a central parameter server aggregates gradient updates in federated learning, or a platform operator brokers compute resources. This architecture fails intelligent networks for three structurally distinct reasons.

Single points of failure. A centralized aggregator is both a performance bottleneck and an existential vulnerability. If the aggregation server is compromised, delayed, or economically captured by a dominant stakeholder, the entire network's epistemic output—the shared model, the aggregated sensory state—can be corrupted or halted. In safety-critical intelligent networks (autonomous vehicle coordination, industrial control systems), this single point of failure is unacceptable from both a reliability and an adversarial-resilience standpoint.

Data silos and epistemic fragmentation. Centralized architectures create strong incentives for participants to withhold or degrade the fidelity of shared data, since contribution is typically decoupled from reward: a node that contributes high-quality data and a node that contributes negligible or low-quality data are frequently compensated identically, or worse, contribution is not compensated at all and is instead extracted as a condition of platform access. This produces a systemic under-provisioning of the very resource—high-quality distributed data and compute—that gives the network its value.

Trust asymmetry. In a centralized model, all nodes must extend unconditional trust to the central operator regarding data usage, model weighting, and reward distribution, while the operator need not symmetrically trust any individual node. This asymmetry is economically unstable in multi-stakeholder settings where participants are commercial competitors, geopolitical rivals, or simply mutually distrusting autonomous agents with no prior relationship. Absent a mechanism for verifiable, trust-minimized coordination, rational nodes will systematically under-invest in the network.

The Role of Tokenization Tokenization addresses this coordination failure by replacing implicit, trust-dependent contribution norms with an explicit, cryptographically enforced, and economically incentive-compatible accounting layer. Formally, we define a token τ within an intelligent network as a cryptographically unique, transferable unit of account whose issuance, transfer, and destruction are governed by a decentralized ledger's consensus rules rather than by any single administrative authority. Tokenization performs three foundational functions simultaneously.

First, it enables automated resource allocation by converting scarce network resources—compute cycles, bandwidth, sensor access, model inference calls—into fungible or non-fungible claims that can be priced, exchanged, and algorithmically rationed via smart-contract-encoded market mechanisms, obviating the need for a human-mediated allocation process.

Second, it enables trust orchestration: because token issuance and transfer are recorded on a tamper-evident distributed ledger and are typically contingent on cryptographically verifiable proofs of work, stake, or contribution, nodes need not trust one another's stated intentions—they need only trust the protocol's verification logic, which is public, auditable, and (in well-

designed systems) deterministic. Third, it enables value transfer without human intervention, which is the property that most distinguishes tokenized intelligent networks from prior distributed-systems architectures. Autonomous AI agents can autonomously earn, spend, stake, and forfeit tokens as a direct consequence of their observed behavior, closing the loop between an agent's actions and its economic consequences at machine speed, without requiring a human operator to approve each transaction.

12.2. THE SECURE INTELLIGENT ECOSYSTEM ARCHITECTURE

A reference architecture for tokenized intelligent networks organized into three interdependent layers. The Physical/Edge Layer comprises the heterogeneous population of sensors, embedded AI inference agents, and edge computation nodes that generate the network's raw informational and computational substrate. This layer is characterized by extreme heterogeneity in computational capability (from milliwatt-class microcontrollers to GPU-equipped edge servers), by intermittent connectivity, and by the fact that the data and compute generated here are the network's fundamental economic commodities. Edge nodes at this layer are typically equipped with lightweight cryptographic identities (public/private key pairs) that allow their contributions to be attributed and subsequently rewarded, and increasingly with hardware roots of trust (secure enclaves, Trusted Platform Modules) that allow contribution claims to be verified even under adversarial conditions. The Blockchain/Consensus Layer is the coordination substrate. It hosts the smart contracts that define token minting and burning rules, staking and slashing logic, and the state channels or off-chain scaling constructs that allow high-frequency micro-transactions (e.g., per-inference micropayments) to occur without incurring the latency and cost of full on-chain settlement for every interaction. This layer's central design tension is between throughput (intelligent networks generate transaction volumes far exceeding what base-layer blockchains can settle directly) and the security guarantees that justify using a blockchain in the first place; this tension is typically resolved through a combination of layer-2 rollups, payment/state channels, and periodic on-chain checkpointing of off-chain state. The Intelligence/Application Layer sits atop the consensus substrate and hosts the higher-order logic that gives the network its purpose: collaborative learning protocols (federated learning aggregators, swarm-consensus algorithms), zero-knowledge proof circuits that allow claims about model performance or data quality to be verified without revealing underlying proprietary data, and automated marketplaces that match compute/data/inference supply with demand via algorithmic price discovery. This layer is where the game-theoretic mechanisms detailed in Section 2 are operationalized as concrete smart-contract logic.

Together, these three layers form a closed economic loop: the edge layer generates value (data, compute, sensing), the consensus layer verifiably attributes and compensates that value via tokens, and the intelligence layer consumes the aggregated output (models, decisions, marketplace allocations) to produce network-level utility that ultimately justifies continued participation by edge-layer nodes. It is this closed loop—rather than any single component in isolation—that constitutes the "secure intelligent ecosystem," and the remainder of this chapter is devoted to formalizing the incentive mechanisms that keep the loop economically stable under conditions of rational self-interest and active adversarial pressure.

12.3. GAME-THEORETIC FOUNDATIONS OF INCENTIVE DESIGN

12.3.1. Mechanism Design in Multi-Agent Systems

The design of tokenized incentives for intelligent networks is, at its mathematical core, an exercise in mechanism design: the inverse problem of game theory, in which the designer specifies the rules of a game—the strategy space, the payoff structure, the information disclosure requirements—such that self-interested agents, each pursuing their own private utility, produce a socially desirable aggregate outcome as an equilibrium of their strategic interaction. We model an intelligent network as a Bayesian game $G = (N, \Theta, A, u, p)$, where $N = \{1, \dots, n\}$ is the set of participating nodes (sensors, edge agents, validators); $\Theta = \Theta_1 \times \cdots \times \Theta_n$ is the space of private types, where $\theta_i \in \Theta_i$ encodes node i's privately known characteristics—its true data quality, its actual computational cost function, or its sensing noise profile; $A = A_1 \times \cdots \times A_n$ is the joint action space, where $a_i \in A_i$ represents node i's observable strategic choice (e.g., how much compute to contribute, what data quality to report, whether to validate honestly); $u_i: A \times \Theta_i \to R$ is node i's utility function, typically decomposed as

$$u_i(a, \theta_i) = R_i(a) - C_i(a_i, \theta_i) \qquad (1)$$

where $R_i(a)$ is the token reward node i receives as a function of the joint action profile (as determined by the smart-contract reward function), and $C_i(a_i, \theta_i)$ is node i's true cost of taking action a_i given its private type θ_i; and $p(\theta)$ is the common-prior probability distribution over the joint type space, representing each node's beliefs about the likely types of its peers.

The designer's task is to specify $R_i(\cdot)$—the token-reward mapping encoded in the smart contract's logic—such that the resulting game admits an equilibrium in which the network's aggregate objective (e.g., maximizing total useful data/compute contributed, or minimizing the variance of a jointly trained model) is achieved, ideally regardless of the specific realization of private types θ.

12.3.2. Key Equilibrium Concepts

Three equilibrium concepts are of particular importance in the design of tokenized incentive mechanisms for autonomous, repeated-interaction networks.

Nash Equilibrium (NE). A strategy profile $a^* = (a_1^*, \dots, a_n^*)$ is a Nash Equilibrium if no node can unilaterally improve its expected utility by deviating, holding all other nodes' strategies fixed:

$$E_{\theta_{-i}}[u_i(a_i^*, a_{-i}^*, \theta_i)] \geq E_{\theta_{-i}}[u_i(a_i', a_{-i}^*, \theta_i)] \quad \forall a_i' \in A_i, \ \forall i \in N \qquad (2)$$

Nash Equilibrium is a comparatively weak solution concept for tokenized intelligent networks because it presumes each node has accurate beliefs about the equilibrium strategies of its peers—an assumption that is fragile in open, permissionless networks with continuous node churn and no guarantee of common knowledge of the mechanism's parameters.

Dominant Strategy Incentive Compatibility (DSIC). A mechanism is DSIC if each node's optimal action is invariant to the actions chosen by every other node—that is, truthful or honest behavior is a best response *regardless* of what other nodes do:

$$u_i(a_i^*, a_{-i}, \theta_i) \geq u_i(a_i', a_{-i}, \theta_i) \quad \forall a_i' \in A_i, \ \forall a_{-i} \in A_{-i}, \ \forall i \in N \qquad (3)$$

DSIC is the strongest and most robust solution concept available to a mechanism designer because it eliminates the need for any node to model, predict, or coordinate with the strategies of its peers—a property of substantial practical importance in intelligent networks where nodes are autonomous software agents with limited capacity for sophisticated strategic

reasoning about the behavior of other agents. Token-reward mechanisms based on Vickrey-Clarke-Groves (VCG) payment rules are the canonical DSIC construction: each node is paid its marginal contribution to social welfare, computed as-

$$R_i(a) = W_{-i}\left(a_{-i}^{opt}\right) - W_{-i}(a) \qquad (4)$$

where $W_{-i}(\cdot)$ denotes the aggregate welfare of all nodes other than i, evaluated once under the allocation that excludes i's report and once under the actual allocation. This payment structure makes truthful reporting a dominant strategy because a node's own report affects only the allocation, not its own marginal-contribution-based payment.

Strict Strategy Proofness. A refinement of DSIC applicable to environments with continuous type spaces (e.g., continuous data-quality or compute-contribution reporting), strict strategy proofness requires that truthful reporting strictly (not merely weakly) dominates any misreport, which is essential in tokenized networks to prevent equilibrium multiplicity in which both honest and marginally dishonest reporting yield identical expected payoffs—an ambiguity that adversarial nodes can exploit probabilistically over repeated interactions even when it confers no advantage in any single round.

12.3.3. The Free-Rider and Malicious Actor Problems

Formalizing the Free-Rider Problem. In a naively designed intelligent network, the shared output (a trained model, an aggregated sensory map) is a non-excludable, non-rivalrous public good: once produced, it can be consumed by any node regardless of that node's contribution, and one node's consumption does not diminish another's. Formally, let $c_i \geq 0$ denote node i's contribution (data volume, compute-hours, sensing accuracy) and let $V(\sum_i^n c_i)$ denote the value of the resulting shared intelligence artifact, which is increasing and typically concave in aggregate contribution. If every node receives access to the artifact regardless of c_i, each node's dominant strategy is $c_i = 0$: a node maximizes its utility $u_i = V(\sum_i^n c_i) - \kappa_i c_i$ (where κ_i is node i's marginal cost of contribution) by free-riding on the contributions of others, since $\partial u_i / \partial c_i = -\kappa_i < 0$ whenever the node treats as fixed. The resulting Nash Equilibrium under free access is $c_i^* = 0\ \forall i$, a catastrophic under-provisioning relative to the socially optimal contribution level, which would maximize $V(\sum_i^n c_i) - \sum_i^n \kappa_i c_i$.

Token-gated access resolves this failure by making the shared artifact excludable: access to the trained model or aggregated intelligence output is conditioned on possessing tokens that can only be acquired by contributing (or by purchasing from contributors on a secondary market, which still requires that someone in the network contributed). This converts the public-goods game into an excludable-club-goods game in which $c_i = 0$ yields $u_i = 0$, restoring a positive-contribution equilibrium.

Byzantine Behaviors and Crypto-economic Countermeasures. Beyond passive free-riding, intelligent networks face active Byzantine adversaries who seek to extract token rewards through misrepresentation rather than mere non-participation. Two behaviors are of particular concern:

Sybil attacks, in which a single adversarial entity creates multiple pseudonymous node identities to disproportionately claim rewards, manipulate reputation-weighted aggregation, or bias validator-selection lotteries. The standard cryptoeconomic countermeasure is to make identity creation costly in a way that is commensurate with the potential reward extraction—

typically via a staking deposit requirement, whereby each node identity must lock σ tokens as a precondition for participation. This transforms the Sybil attacker's problem from "create k identities at near-zero cost" to "acquire and lock $k\sigma$ tokens," directly linking Sybil resistance to the attacker's capital constraints.

***Data poisoning in federated learning*,** in which a malicious node submits crafted model updates designed to degrade the global model's performance or to embed a backdoor, while still appearing to comply with the protocol's surface-level requirements. Because data poisoning is difficult to detect from a single update in isolation, token mechanics address it through a combination of staking (requiring nodes to post collateral σ_i that is forfeited if their contribution is later proven, via cross-validation against a held-out reference set or via peer-audit, to be malicious) and slashing (the protocol-enforced burning or redistribution of a fraction $\lambda \in (0,1]$ of a node's staked collateral upon detected misbehavior). The expected utility of a poisoning attack becomes

$$E\left[u_i^{attack}\right] = \pi_{success} \cdot G_{attack} - \pi_{detect} \cdot \lambda\sigma_i \qquad (5)$$

where $\pi_{success}$ is the probability, the poisoned update evades detection and achieves the adversary's objective, G_{attack} is the adversary's gain from a successful attack, and π_{detect} is the probability of detection and slashing. A well-calibrated mechanism sets σ_i and λ such that $E\left[u_i^{attack}\right] < E\left[u_i^{honest}\right]$ for all rational adversaries across the plausible range of G_{attack}, shifting the utility balance decisively toward honest participation as the only economically rational equilibrium strategy—the central cryptoeconomic design objective of Sections 3 and 4.

12.4. TOKENOMICS TAXONOMY FOR INTELLIGENT RESOURCE ALLOCATION

12.4.1. Token Classification

Tokenized intelligent networks typically deploy multiple, functionally distinct token classes within a single protocol, each engineered to solve a different incentive-alignment problem. Conflating these classes—a common design error in early tokenized-IoT projects—produces mechanisms in which governance power, economic access rights, and security collateral are fungible with one another, creating attack surfaces in which an adversary can, for instance, purchase governance influence using tokens earned through low-effort resource provisioning rather than through demonstrated protocol commitment.

Utility Tokens. A utility token τ_U functions as the network's medium of exchange for accessing metered resources: compute cycles, API calls to a deployed inference endpoint, bandwidth for data transmission, or storage. Its economic property of interest is a stable, demand-responsive exchange value against the resources it gates; formally, the equilibrium price of a utility token can be modeled via a variant of the equation of exchange,

$$P_U \cdot V = M \cdot Q \qquad (6)$$

where P_U is the token's price, V is its velocity (transactions per unit time), M is the circulating supply, and Q is the aggregate real value of resources transacted through the token per unit time. High velocity—tokens acquired and immediately spent, as is common in pure access-utility designs—tends to suppress the token's price appreciation relative to network growth, a well-documented tension in utility-token design sometimes termed the "velocity problem," which motivates the staking and burning mechanisms discussed below.

Governance Tokens. A governance token τ_G confers voting rights over protocol parameters: fee schedules, staking requirements, slashing severity λ, and admission criteria for new resource providers. Governance tokens are typically non-consumptive (holding them does not, by itself, grant resource access) and their value derives from the option value of influencing future protocol evolution. In intelligent networks specifically, governance token design must address the risk of plutocratic capture, in which entities controlling large compute or data resources also accumulate disproportionate governance weight and subsequently vote to adjust parameters in their own favor (e.g., lowering staking requirements to reduce their own capital lockup). Quadratic voting schemes, in which voting power scales with $\sqrt{\tau_G}$ rather than linearly with τ_G, and reputation-weighted governance overlays (Section 5.2) are common mitigations.
Staking/Security Tokens. A staking token τ_S—which may be the same underlying asset as τ_U or a distinct class—serves as collateral that validators, data contributors, or compute providers lock as a bond against honest behavior, subject to slashing per the mechanism of Section 2.3. The security guarantee a staking mechanism provides scales with the aggregate value at stake relative to the potential gain from misbehavior; formally, the network's Byzantine fault tolerance threshold is economically meaningful only if

$$\sum_{i}^{n} \sigma_i \cdot \lambda \; > \; \max_{\text{coalition } C} G_{attack}(C) \qquad (7)$$

for all feasible malicious coalitions C, i.e., the total slashable value exceeds the maximum extractable gain from any coordinated attack.

12.4.2. Dynamic Supply Dynamics

Static token supply schedules are poorly suited to intelligent networks whose resource demand fluctuates with sensing conditions, model retraining cycles, and compute-market cyclicality. Two dynamic supply mechanisms are of particular relevance.
Minting tied to verified contribution. New tokens are minted as a direct function of verified network output—e.g., a fixed emission μ tokens per unit of cryptographically attested compute delivered, or per unit of Shapley-value-weighted data contribution (Section 4.2). This ties monetary expansion directly to real economic throughput rather than to time, avoiding the inflationary dilution that plagues fixed-schedule emission models during periods of low network utilization. Algorithmic fee-burning. Adapting the EIP-1559 base-fee-burning mechanism to resource-constrained intelligent networks, each resource-access transaction (an inference call, a data-stream subscription) pays a base fee f_{base} that is algorithmically adjusted block-to-block (or epoch-to-epoch) as a function of realized network utilization u_t relative to a target utilization u^*:

$$f_{base}^{t+1} = f_{base}^{t} \cdot \left(1 + \delta \cdot \frac{u_t - u^*}{u^*}\right) \qquad (8)$$

where δ is a damping constant bounding the fee's adjustment rate per epoch. The base fee is burned (permanently removed from circulating supply) rather than paid to any party, decoupling network congestion pricing from validator/provider rent-extraction and instead using it purely as a supply-elasticity valve: sustained high demand burns tokens faster than they are minted, appreciating the token for holders and providers even as access becomes more expensive for marginal consumers, which is the intended discipline against resource over-consumption in constrained edge environments.

12.4.3. Resource Monetization Models

Compute Provisioning. Tokenized compute marketplaces—architecturally similar to Akash Network's reverse-auction compute marketplace or Render Network's GPU-rendering marketplace—operate as continuous double auctions in which compute *providers* post ask orders specifying available capacity (CPU/GPU-hours, memory, specialized accelerators) and a reservation price, while compute *demanders* (e.g., an edge-AI agent needing to offload a training job) post bid orders specifying required capacity and a maximum willingness to pay. A smart-contract-resident matching engine clears the market at each epoch, typically via a uniform-price or pay-as-bid auction, and escrows tokens from the demander until the provider furnishes a cryptographically verifiable proof of computation (e.g., a succinct proof that the claimed computation was performed correctly, or in weaker designs, redundant execution with majority-vote verification across multiple providers).

Data Supply Chains. Tokenized data marketplaces extend this model to raw or processed sensor/training data, with the crucial addition of a data provenance tracker: each data unit is registered on-chain with a cryptographic hash, a timestamp, and an attributable source identity, allowing downstream consumers to verify data lineage and allowing upstream contributors to receive royalty-style token flows whenever their registered data is subsequently used to train a model that generates further downstream value—a design pattern sometimes termed "data dividends," implemented via smart contracts that track derivative usage and distribute a fraction ρ of downstream model-usage revenue back to the original data contributors in proportion to their (Shapley-value-weighted) contribution to model performance.

Model-as-a-Service (MaaS). Decentralized inference marketplaces allow trained models to be deployed across a distributed network of inference-serving nodes, with clients paying per-inference-call token micropayments (typically settled via state channels to avoid on-chain latency for each call) and inference-serving nodes competing on latency, cost, and—critically for trust-minimized deployment—verifiability. Verification in decentralized MaaS settings typically relies on one of two approaches: optimistic verification, in which inference results are accepted by default but can be challenged within a dispute window by any party willing to post a bond and re-run the computation, with the loser of the dispute forfeiting their bond; or zero-knowledge verified inference, in which the serving node produces a succinct cryptographic proof that the claimed model was executed correctly on the claimed input, allowing the client to verify correctness without re-execution and without the serving node revealing proprietary model weights—a computationally expensive but rapidly maturing approach as zero-knowledge machine-learning (zkML) proving systems improve in efficiency.

12.5. CASE STUDY PROTOCOL: INCENTIVIZING COLLABORATIVE INTELLIGENCE

12.5.1. Decentralized Federated Learning (DFL) Architecture

We now specify a concrete tokenized protocol for Decentralized Federated Learning that operationalizes the mechanisms of Sections 2 and 3. The architecture replaces the single trusted parameter server of classical federated learning with a blockchain-coordinated aggregation process, structured as follows.

Each participating node i maintains a local dataset D_i (never transmitted off-device) and, at the start of each training round t, downloads the current global model parameters w_t from a

content-addressed off-chain storage layer (e.g., IPFS), with the content hash anchored on-chain for tamper-evidence. Node i performs E local epochs of gradient descent on D_i, producing an updated local model w_i^{t+1}. Rather than transmitting raw gradients directly on-chain (which would be prohibitively expensive), node i submits: (1) the content hash of w_i^{t+1} to off-chain storage; (2) a commitment to a small validation-set performance metric (accuracy or loss delta) computed against a held-out reference set whose composition is unknown to node i at training time, preventing overfitting to the validation criterion; and (3) a cryptographic signature binding the submission to node i's on-chain identity and staked collateral σ_i.

Aggregation occurs via a state channel among a rotating committee of validator nodes selected by verifiable random function (Section 5.2), who collect the round's submissions, compute a stake-and-quality-weighted federated average.

$$w_{t+1} = \sum_{i \in R_t}^{n} \phi_i \cdot w_i^{t+1}, \qquad \phi_i = \frac{q_i \cdot \sigma_i}{\sum_{j \in R_t}^{n} q_j \cdot \sigma_j} \tag{9}$$

(where R_t is the set of nodes participating in round t and q_i is node i's quality score from Section 4.2), and commit the resulting aggregate hash and the round's reward/slashing determinations to the base-layer blockchain via a single settlement transaction, amortizing on-chain gas costs across the entire committee's round of activity.

Proof of Contribution (PoC) Mechanisms

The central technical challenge in tokenized DFL is verifying, in a way that is both computationally tractable and resistant to gaming, how much a given node's contribution improved (or degraded) the shared model—without requiring the network to trust the contributing node's self-reported metrics.

Shapley Value Estimation for Data Quality Valuation. The Shapley value, adapted from cooperative game theory, provides a principled, axiomatically fair allocation of credit among contributors to a joint outcome. For node i's data contribution, its Shapley value is

$$SV_i = \sum_{S \subseteq R_t \setminus \{i\}}^{n} \frac{|S|!\ (|R_t| - |S| - 1)!}{|R_t|!} [V(S \cup \{i\}) - V(S)] \tag{10}$$

where $V(S)$ denotes the validation-set performance (e.g., negative loss) of a model trained using only the data contributions of coalition S. This formulation credits node i with the average marginal improvement its data provides across all possible orderings of the other contributors, which correctly accounts for complementarities and redundancies between contributors' data (e.g., two nodes with near-duplicate datasets should each receive diminished marginal credit relative to a node contributing a novel data distribution). Because exact Shapley value computation is combinatorially expensive ($O\left(2^{|R_t|}\right)$ coalition evaluations), practical DFL protocols employ Monte Carlo Shapley estimation, sampling a bounded number K of random permutations of R_t and computing the average marginal contribution across the sampled permutations, which converges to the true Shapley value with estimation error decreasing as $O\left(1/\sqrt{K}\right)$—a tractable trade-off between committee computational overhead and valuation precision that can itself be parameterized via governance.

Zero-Knowledge Proofs for Accuracy Verification. To avoid requiring the validation committee to access raw training data (which would violate the privacy guarantees that motivate federated learning in the first place) or even to trust a node's self-reported local-training procedure, zkML circuits allow a contributing node to produce a succinct proof π_i attesting that: (a) the

submitted model update w_i^{t+1} was in fact derived by running the agreed-upon training algorithm (specific optimizer, learning rate, epoch count) starting from w_t on *some* dataset satisfying declared size and schema constraints; and (b) the claimed validation-performance delta was computed correctly against the committee-issued validation set. The validating committee can verify π_i in time sublinear in the cost of the original computation, without observing D_i or the intermediate gradients, closing the primary avenue for a node to fabricate its contribution metrics while still preserving the confidentiality of proprietary local data.

12.5.2. Reward and Slashing Algorithms

Algorithm 1 specifies the smart-contract reward and slashing procedure executed by the validating committee at the conclusion of each training round t.

Algorithm 1. RoundSettlement $\left(\boldsymbol{R_t}, \{\boldsymbol{w_i}, \boldsymbol{\pi_i}, \boldsymbol{\sigma_i}\}_{\boldsymbol{i \in R_t}}, \boldsymbol{V_{ref}}\right)$

```
 1: for each node i in R_t:
 2:     valid_i <- VerifyZKProof(pi_i, w_i, V_ref)
 3:     if valid_i = false:
 4:         Slash(i, lambda_fraud * sigma_i)      // proof-invalid =>
presumptive fraud
 5:         Emit(SlashedInvalidProof, i)
 6:         continue
 7:     q_i <- MonteCarloShapley(i, R_t, V_ref, K)
 8:     if q_i < q_min:                           // contribution below
minimum threshold
 9:         Slash(i, lambda_poor * sigma_i)
10:         Emit(SlashedLowQuality, i)
11:         continue
12:     if q_i < 0 and |q_i| > tau_poison:         // likely data
poisoning
13:         Slash(i, lambda_poison * sigma_i)
14:         Emit(SlashedPoisoning, i)
15:         continue
16:     R_i <- mu * (q_i / sum_j q_j) * TotalRoundEmission   // node
passes verification
17:     Mint(i, R_i)
18:     Emit(RewardIssued, i, R_i)
19: end for
20: w_{t+1} <- WeightedAggregate({w_i, q_i, sigma_i} for i where valid_i
and q_i >= q_min)
21: CommitToChain(Hash(w_{t+1}), RoundSummary)
```

This settlement procedure enforces three distinct incentive gradients simultaneously. Nodes submitting invalid or unverifiable proofs face immediate, severe slashing (λ_{fraud}, calibrated to exceed any plausible gain from proof forgery), independent of the actual quality of their underlying data, since an unverifiable claim cannot be distinguished from a fabricated one and must be treated as such for the mechanism to remain incentive-compatible. Nodes submitting valid but low-quality contributions face milder, threshold-triggered slashing (λ_{poor}) intended to discourage negligent or lazy participation without over-penalizing honest nodes whose local data happens to be less informative through no fault of intent. Nodes whose contributions are

verified to have a statistically significant *negative* marginal effect on model performance—the signature of a data-poisoning attack—face the most severe slashing tier (λ_{poison}), calibrated per the inequality in Equation [eq:bft] to exceed the maximum plausible gain from a successful poisoning attack across the full space of adversarial objectives the network anticipates (targeted misclassification, backdoor insertion, general model degradation). Honest, high-quality contributors receive newly minted tokens in direct proportion to their Shapley-value share of the round's aggregate quality, tying monetary reward precisely to verified marginal contribution rather than to raw participation volume, and thereby resolving the free-rider problem formalized in Section 2.3 while simultaneously making honest participation—rather than any form of misreporting—the unique dominant strategy under the DSIC-approximating reward structure this algorithm implements.

12.6. SECURITY, SYBIL RESISTANCE, AND ATTACK VECTORS

12.6.1 Cryptoeconomic Attack Vectors

Tokenized intelligent networks inherit the full attack surface of the underlying blockchain layer while simultaneously introducing novel attack vectors specific to the intelligence-generation process they coordinate. A rigorous security analysis must treat these as compounding rather than independent risks. 51% Exploits and Long-Range Attacks on Intelligence-Led Consensus. Where the consensus mechanism securing the ledger is itself influenced by intelligence-network-specific metrics—for instance, a Proof-of-Contribution variant in which validator selection probability is weighted by historically accrued reputation or Shapley-value contribution rather than by pure stake—an adversary controlling a supermajority of either the underlying stake *or* a supermajority of accumulated reputation can bias validator committee selection to systematically favor colluding validators. This is structurally analogous to a classical 51% attack but with an expanded attack surface: the adversary need not necessarily acquire 51% of token supply if reputation-weighting can be gamed through a long-range strategy of initially honest participation designed to accrue disproportionate reputation before pivoting to exploitation—a variant we term a reputation long-range attack, in which the "long range" is temporal accumulation of trust rather than the classical long-range attack's rewriting of historical chain state.

Collusion Rings. In committee-based validation architectures (Section 4.1), a coalition of edge nodes may coordinate off-protocol to submit mutually corroborating but substantively fraudulent validation metrics—for example, a ring of nodes that all attest to each other's low-quality or even non-existent data contributions as high-quality, extracting token rewards without commensurate real contribution. Formally, if a colluding coalition C controls a sufficient fraction of the validating committee in a given round (i.e., $|C \cap R_t|/|R_t|$ exceeds the honest-majority assumption threshold), the coalition can manipulate the reported Shapley-value estimates q_i for its members, since Monte Carlo Shapley estimation as specified in Section 4.2 relies on the validation set and computation being performed honestly by the committee. The economic feasibility of such collusion is governed by comparing the coalition's expected extracted value to the aggregate stake at risk across the colluding members, which the slashing calibration inequality of Equation [eq:bft] must account for under a coalition-adjusted threat model rather than assuming only independent, uncoordinated adversaries.

Bribery and MEV in Tokenized Data Marketplaces. Validators or committee members responsible for ordering, matching, or settling transactions in tokenized compute/data/inference marketplaces (Section 3.3) possess Maximal Extractable Value: the ability to profit by reordering, inserting, or censoring transactions ahead of their execution. In an intelligent-network data marketplace, this manifests as validators front-running favorable data-purchase orders, sandwiching automated market-maker-based compute-price transactions, or selectively censoring competing bids from rival compute providers in whom the validator has an undisclosed economic interest. A distinct but related threat is bribery attacks, in which an external adversary pays validators (in tokens, potentially even the network's own native token, creating a reflexive attack economy) to bias their validation decisions—for instance, bribing committee members to approve a poisoned model update or to under-report a competitor's Shapley-value contribution.

12.6.2 Mitigation Matrix

Token Curation Registries (TCRs). To mitigate the risk of low-quality or adversarial nodes gaining committee eligibility, intelligent networks can employ token-curated admission lists in which existing high-reputation stakeholders vote (with their own tokens at implicit risk, since the TCR's overall quality reflects on the value of all participants' holdings) on the admission of new validator or high-privilege node identities, creating an economically self-reinforcing quality filter that operates orthogonally to pure stake-weighting and is therefore resistant to attacks that rely solely on capital acquisition.

Reputation-Weighted Voting. Rather than weighting governance and committee-selection influence purely by token stake (which is vulnerable to plutocratic capture and to capital-rich short-term adversaries), reputation-weighted schemes incorporate a decaying historical-performance metric

$$\rho_i^t = \gamma \rho_i^{t-1} + (1-\gamma) q_i^t \qquad (11)$$

(an exponentially weighted moving average of a node's historical Shapley-value contributions, with decay factor γ controlling the "memory" of the reputation system), such that influence must be earned through sustained honest contribution over time and cannot be instantaneously purchased—directly countering the reputation long-range attack described above by making the accumulation of exploitable reputation costly in both time and consistent honest behavior, which raises the effective cost of the attack beyond what a purely capital-based analysis would suggest. Verifiable Random Functions for Validator Selection. To prevent an adversary from predicting or manipulating which specific nodes will constitute a given round's validation committee (a prerequisite for both collusion-ring formation and targeted bribery), committee selection should be performed via a VRF: each eligible node computes –

$$(output, \pi) = VRF_{sk_i}(seed_t) \qquad (12)$$

using its private key and the current round's public seed (itself derived unpredictably from prior block hashes), producing a pseudorandom output that determines committee eligibility and that can be publicly verified using the node's public key without revealing the private key or allowing the output to be predicted in advance of computation. This ensures committee composition is unpredictable until the moment of selection, eliminating the pre-round bribery and collusion-targeting window that would exist under any deterministic or early-announced selection scheme. Trusted Execution Environments Combined with Tokens. For the subset of

computation where zero-knowledge proof generation remains computationally impractical (particularly for large-model training rather than inference), hardware-enforced security via Trusted Execution Environments (TEEs)—secure enclaves that provide cryptographic attestation that a specific piece of code executed correctly and untampered on a specific piece of hardware—offers a complementary verification path. Token mechanics integrate with TEE attestation by conditioning reward eligibility on the presentation of a valid TEE attestation alongside the standard proof-of-contribution artifacts, and by staking additional collateral against the specific hardware attestation identity, creating a dual verification layer (cryptographic-hardware plus cryptoeconomic-stake) whose combined security exceeds either mechanism in isolation—critical given that TEEs, while offering strong practical guarantees, are not immune to side-channel and physical-access vulnerabilities, and should therefore be treated as risk-reducing rather than risk-eliminating within the broader cryptoeconomic security model.

12.7. FUTURE HORIZONS AND CONCLUSION

12.7.1. Cross-Chain Interoperability for Swarm Intelligence

As intelligent networks proliferate across independently governed blockchain ecosystems—an autonomous vehicle fleet coordinating on one chain, an environmental sensor swarm on another, a federated healthcare-diagnostics consortium on a third—the value of cross-network resource exchange grows correspondingly, motivating a shift from siloed, chain-specific tokenized intelligent networks toward genuinely interoperable swarm intelligence architectures. Cross-chain messaging protocols such as the Inter-Blockchain Communication protocol (IBC) and generalized cross-chain messaging layers such as LayerZero provide the technical substrate for this interoperability, allowing a smart contract on one chain to trigger verified state changes on another—for instance, allowing a compute-rich intelligent network on one chain to sell surplus GPU capacity to a data-rich but compute-poor sensor network on another chain, with payment and delivery-confirmation messages relayed and cryptographically verified across the chain boundary. The core technical challenge in extending the incentive mechanisms of Sections 2 through 5 to a cross-chain setting is that the strong assumptions underpinning single-chain mechanism design—synchronous finality, a shared validator set with aligned slashing exposure, a single canonical token whose supply dynamics are unambiguous—no longer hold. A cross-chain Sybil or collusion attack can potentially exploit asynchronies between chains' finality guarantees (a variant of a cross-domain double-spend), and slashing collateral posted on one chain may not be efficiently seizable to compensate for damage inflicted on a counterparty operating on a different chain, absent a cross-chain collateral-escrow mechanism. Emerging designs address this via cross-chain restaking, in which a node's stake on a base security-providing chain is cryptographically committed to back its behavior across multiple downstream intelligent-network deployments simultaneously, allowing a unified security budget to be shared across a heterogeneous, multi-chain swarm rather than requiring each chain-specific deployment to bootstrap independent, and therefore individually weaker, cryptoeconomic security.

12.7.2. AI-Driven Algorithmic Tokenomics

A further horizon, still largely at the research-prototype stage at the time of writing, is the delegation of the tokenomic parameter-tuning problem itself to an AI agent operating within the network it governs. Rather than fixing emission rates μ, staking requirements σ, slashing severities λ, or fee-burn damping constants δ via static governance votes—which are slow relative to the timescale of network-state fluctuation and are themselves subject to the plutocratic-capture risks discussed in Section 5.2—a self-optimizing tokenomic agent could continuously observe network-health telemetry (validator participation rates, realized attack attempts, token velocity, resource-utilization variance) and dynamically adjust these parameters within governance-bounded ranges to maintain target security and liquidity conditions. This approach raises a distinctive second-order incentive-design problem: the AI agent adjusting the network's tokenomics is itself an economic actor whose objective function must be specified and whose potential for misalignment (whether through optimization for a poorly specified proxy metric, or through adversarial manipulation of the telemetry it observes) must be bounded by the same mechanism-design rigor applied to the human-operated nodes elsewhere in the network. Plausible mitigations mirror those already established in this chapter: the tokenomic agent's parameter-adjustment authority could itself be staked and slashable against verifiable network-health outcomes, and its adjustment range could be bounded by governance-set guardrails (a maximum per-epoch percentage change in any given parameter) that prevent even a fully compromised agent from making catastrophic, network-destabilizing adjustments within any single decision cycle—an application of the general cryptoeconomic principle, established throughout this chapter, that autonomous decision-making authority within a tokenized intelligent network should be granted only in proportion to the collateral or reputational stake that authority's holder has placed at risk.

This chapter has argued, and sought to substantiate through concrete architectural, game-theoretic, and algorithmic detail, that tokenized incentive mechanisms are not a peripheral monetization layer bolted onto otherwise-complete intelligent network architectures, but rather constitute the critical infrastructure without which autonomous, multi-stakeholder intelligent coordination is not economically stable. The free-rider and Byzantine-actor problems formalized in Section 2 are not incidental engineering challenges but structural consequences of coordinating self-interested, privately-informed autonomous agents absent a central trusted authority; left unaddressed, they drive rational nodes toward non-participation or active exploitation, unraveling the very collective intelligence the network exists to produce. The tokenomics taxonomy of Section 3, the Proof-of-Contribution protocol of Section 4, and the security mitigation matrix of Section 5 collectively demonstrate that these structural problems admit tractable, cryptographically and economically rigorous solutions—provided the mechanism designer treats incentive compatibility, Sybil resistance, and verifiable contribution attribution as first-order architectural requirements rather than downstream additions. As intelligent networks continue their trajectory toward greater autonomy—more decision-making authority delegated to AI agents operating with less direct human oversight, across increasingly heterogeneous and cross-chain deployments—the rigor with which their underlying tokenized incentive mechanisms are designed will directly determine whether these networks remain secure, collaborative, and self-sustaining, or whether they degrade into extractive equilibria dominated by the very free-riding and Byzantine behaviors this chapter has sought to formally characterize and cryptoeconomically neutralize. The synthesis of mechanism design, applied cryptography, and distributed-systems engineering presented here

is offered as a foundation for that continuing design effort, rather than as a finished prescription; the specific parameterizations, proof systems, and cross-chain protocols will necessarily evolve, but the underlying game-theoretic imperative—that autonomous economic behavior within intelligent networks must be structured so that honest, contributive participation is the uniquely rational strategy—will remain the field's central design constraint.

CHAPTER 13

SECURITY CHALLENGES IN BLOCKCHAIN AND AI SYSTEMS

Contributed by: Arnab Das, Dibyani Samanta, Sudeshna Saha, Somenath Goswami

13.1. INTRODUCTION

13.1.1. Background and Context

Undoubtedly, there is a paradigm shift happening in the architecture of modern computing, owing to the rapid advancement of two cutting-edge technologies, viz., block chain and artificial intelligence. In the past, centralized perimeter protected model was the dominant mode of computing. The main concept of centralized architecture was that the central server or data center served as the basis for computing and one point of authority. But in today's hyperconnected world, the traditional centralized models turned out to be the single point of failure, which could not be easily scaled up against sophisticated cyber-attacks. On the other hand, the concept of blockchain computing is based on decentralization and trustless networks. Using a peer-to-peer system, cryptography and consensus protocols, it develops an immutable and transparent ledger. Blockchain allows different entities having varied degree of trust between each other to trade assets and communicate without any middleman. Regardless of whichever type of blockchain model – proof of work, proof of stake or permissioned Byzantine fault tolerant algorithm – is used, the common element of blockchain is mathematical proof of data availability and integrity.

13.1.2 The Convergence of Blockchain and Artificial Intelligence

Statistical intelligence and cryptocurrency work together to produce a strong synergistic effect since the advantages of one technology compensate for the disadvantages of the other. At the elementary level, statistical intelligence is stochastic, dynamic, and opaque, while blockchain technology is deterministic, immutable, and completely traceable. This integration of two different paradigms creates the following three paradigm shifts:

A. Decentralised AI Marketplaces and Crowdsourced Computing

Due to the few IT companies having the big data warehouse needed to train the large models, AI research continues to be quite centralized. Centralized AI gives rise to monopoly problems and data privacy challenges. Blockchain offers disruption by creating distributed AI platforms

such as SingularityNET and Fetch.ai. The data scientists can run their machine learning models on a chain of blocks via smart contracts, where the users get tokens for inference from APIs through a cryptocurrency.

B. Immutable Data Provenance and Model Training Lineage

The main problem in modern conventional machine learning lies in proving data integrity. This is due to the fact that the computer algorithms of learning depend very much on the data used to teach them, and any changes or manipulations made on that data result in a completely unreliable model. Using the blockchain technology, one can solve this problem of tracking because by using the blockchain transaction, the origin certificates of the dataset, metadata, and cryptographic hash of the dataset are recorded before the data set is used in the machine learning algorithm.

C. Autonomous Multi-Agent Systems and Decentralised Autonomous Organisations (DAOs)

The use of self-driving AI bots that are implemented in intelligent contracts through the DAO framework would become the result of this merger. In these highly advanced systems, routine operating management is not performed by people; business rules, budgets, and tokenomics measures are determined by the DAO consensus level, and then an independent AI bot performs the process. Thus, for example, an AI bot would have the opportunity to observe the energy world market, calculate the energy consumption growth based on local climatological data, and then execute smart contracts for buying, storing, and selling energy across the decentralized energy network. Such a network would function as a secure virtual world, where a machine learning agent has the liquidity and logistics abilities, whereas all transactions take place constantly.

13.1.3 Problem Statement

Despite the absence of any doubts regarding the benefits of combining both technologies, the combination of AI with cryptocurrencies becomes highly complex concerning potential attacks. The current approaches to cybersecurity, such as authentication-based NAT firewalls, identity and access management (IAM), and network segmentation, will be completely useless in such circumstances. For basic cybersecurity purposes, it is assumed that there are two types of actors; the actor can either be located inside the credible network with a firewall or outside it. In case of blockchains, this concept is not relevant due to their random nature and the lack of clear borders. A publicly available node in the block network processes all transactions openly, making consensus and smart contracts open to inspection. If there is any flaw in the programming of the contract, it might lead to quick financial losses.

Nevertheless, the unpredictable behavior of algorithms of machine learning (artificial intelligence) is too complex for traditional means of protection to deal with. Neural networks implementing deep learning are considered "black boxes" since they make decisions according to a myriad of parameters (weights) of their inner activity and not in accordance with logical norms understandable for a person. Due to the complexity of their design, the algorithms used in artificial intelligence are vulnerable to attacks since minor alterations in the structure of input data (altering the location of several pixels in an image or in a flow of information) are enough to fool the algorithm. They take advantage of math logic of the network to deceive without changing the structure of the data. The disadvantages of such technologies can become serious only upon their integration into one system. One such disadvantage can be the

predictive finality typical for block chains, which can result in the possibility that mistakes committed by the statistical model of artificial intelligence will be irreversible. If an active attack convinces the AI to make a wrong move, then the regular block chain procedure will record a transaction permanently, with no way for the central operator to reverse it. On the other hand, should the block chain itself or its data source fail, then the data used by AI will be endangered.

13.2. CORE SECURITY CHALLENGES IN BLOCK CHAIN SYSTEMS

The block chain technology has proved itself as an innovative technology that enables the creation of decentralised digital infrastructure without the presence of any authority in its architecture. The base of this technology relies on cryptography and distributed consensus with immutable ledger system that assures transparency and trust among different entities participating in this process. Because of this characteristic, block chain has been applied to many applications ranging from digital currency, decentralized finance (DeFi), healthcare records, supply chain tracking, and digital identity systems. But because of such architecture, the block chain technology faces some security challenges which are distinct from the conventional centralised computing environment. In contrast to traditional information systems, there is no centralized authority that can reverse transactions and recover compromised data in block chain networks. After being verified and recorded in the distributed ledger, transactions cannot be altered or deleted anymore. This way, any vulnerability of the consensus algorithm, smart contract code, or the infrastructure itself may result in irreversible damage such as financial loss and service disruption, thus leading to distrust towards block chain technology. Moreover, since block chains are public networks, the potential adversaries constantly scrutinize their consensus algorithms, transaction history, and smart contracts for any loopholes to exploit. The security problems of block chains can be roughly divided into three categories: protocol/consensus issues, smart contract flaws, and network/infrastructure problems.

13.2.1 Protocol and Consensus Layer Vulnerabilities

The consensus layer provides the backbone of a block chain network through enabling distributed nodes to reach agreement on a shared ledger without using any centralised entity. Consensus algorithms like Proof of Work (PoW) and Proof of Stake (PoS) algorithms ensure data consistency and fault tolerance. Yet, these consensus algorithms continue to be exposed to various protocol-based attacks, which threaten the security of the network. In the following figure, we can see the 51% attack, one of the most important attacks that threatens the block chain network, which happens when a single miner or mining pool attains over 50% computing power in a block chain network. By having majority power, the attacker is able to build its own block chain, perform double spending, stall transaction confirmations, and censor some transactions. Although it is highly challenging for such attacks to take place in a large block chain network due to large computing power needed, smaller block chain networks are much easier to hack. One more critical vulnerability is selfish mining that involves miners intentionally withholding newly found blocks rather than announcing them right away. Through privately growing their own block chain and then making it public, attackers will be able to nullify the efforts of honest miners and get more of mining rewards, thus contributing to the centralisation

of the mining process and decreasing its decentralisation. Blockchain networks are subject to Sybil attacks that involve creation of numerous fake identities/nodes by an attacker to achieve excessive control over the operation of the network. Fake identities/nodes can influence peer discovery process, transactions distribution, and decentralised voting procedures, especially in the case of lightweight/permissioned block chain networks. Though resource-intensive consensus algorithms make Sybil attacks less effective, additional measures should be taken for the verification of identity. In addition, the increasing use of cross-chain communication, sidechains, and Layer-2 approaches has brought about new security issues at the protocol level. Weakness in one interconnected blockchain could lead to vulnerabilities in others that are connected. This way, the attack surface increases. In order for the consensus layer to be secure, there need to be improvements made to consensus algorithms, incentives, and cross-chain security.

13.2.2 Smart Contract Vulnerabilities

Besides protocol and smart contract attacks, the security issues that exist in the block chain systems include those that arise due to security concerns at the network and infrastructure level. In a block chain system, nodes constantly communicate using peer-to-peer networks in order to share transactions and synchronize the distributed ledger. Failure of the communication process may have impacts on the availability, integrity, and dependability of the block chain systems. The eclipse attack poses serious problems because it allows attackers to isolate a node in the block chain by controlling its network traffic. The node becomes vulnerable to the acceptance of malicious transactions because of receiving block chain information from malicious peers. The eclipse attack paves the way for more dangerous attacks including the double spending and selfish mining attacks by isolating nodes from the honest network. Yet another critical issue is key management. The possession of the appropriate private key gives ownership of block chain-based resources, and any entity that has a proper key will gain full access to the relevant assets. In contrast to other systems, it is practically impossible to retrieve lost or stolen keys by going through any centralized organization. Private keys may be accessed through phishing attacks, malware, hacked devices, and social engineering. For this reason, various means of storage, like hardware wallets, multi-sig wallets, and Hardware Security Modules (HSM), have become common practices to lower the chances of unauthorized access.

Finally, block chain technologies are vulnerable to the same issues as conventional cyber security systems are. Distributed Denial-of-Service (DDoS) attacks, software vulnerabilities, as well as failures of other related systems, such as APIs, cloud infrastructure, and wallets, can impact the overall security of block chain platforms.

To summarize, it is necessary to note that it is vital to guarantee the secure communication channels, to maintain an efficient key management system, to perform constant monitoring and security checks. Enhancing these layers will help make block chain networks more reliable and resilient to security threats.

13.2.3 Network and Infrastructure Risks

Along with protocol and smart contract weaknesses, there are various security issues in block chain systems that relate to networks and infrastructures as well. Block chain nodes constantly exchange data via peer-to-peer communication in order to synchronize transactions and block

chain. Interruption of any sort in this communication may have a negative impact on the availability and integrity of the block chain system. The eclipse attack poses one of the greatest threats because it makes it impossible for a block chain node to interact with other nodes of the network due to the fact that all its connections are under control of the hacker. Consequently, the victim node gets its information only from malicious nodes and may accept some fraudulent transaction data or outdated block chain.

A second issue is private key management. Ownership of the block chain resources depends entirely on the use of cryptographic private keys; and any individual holding the right private key will be able to take complete possession of the respective digital resources. In contrast with other systems, private keys cannot usually be restored using any centralised authority if they have been lost or stolen. The attackers typically hack into a private key via phishing, malware infection, exploitation of insecure hardware, and/or social engineering attacks. Thus, using some of the secure storage options such as hardware wallets, multi-signature wallets, and HSMs is highly recommended in order to avoid potential security breaches. The infrastructure of the block chain systems is also susceptible to conventional cyber security threats such as Distributed Denial-of-Service (DDoS) attacks, software vulnerabilities, and failures of the third-party services. Many block chains are built with the help of different APIs, cloud infrastructure and wallet providers; thus, vulnerabilities in these systems can potentially affect the stability of block chains. In general, ensuring the security of block chain infrastructure requires using secure communication over the network, appropriate key management, regular monitoring, and security assessments.

13.3. CORE SECURITY CHALLENGES IN ARTIFICIAL INTELLIGENCE (AI) SYSTEMS

AI has become an integral part of current digital environments facilitating intelligent automation, prediction, and autonomous decisions in various industries such as healthcare, finance, manufacturing, transportation, and cybersecurity. Nonetheless, AI systems represent new security challenges which differ from security concerns related to conventional software systems. The performance of machine learning models depends on the quality of data which is used for their development and use. Thus, attacks targeting AI systems usually involve data, learning process, or a model rather than computing infrastructure itself. Nonetheless, security of AI systems involves not only traditional network and application protection but also requires securing the whole process of machine learning. Machine learning process consists of several stages including data gathering, model training, deployment, inference and continuous model updates. Problems in any of these phases can affect the performance of a model, expose sensitive information, or even enable the manipulation of decision making.

13.3.1 Adversarial Machine Learning

Adversarial machine learning involves the process in which malicious users attempt to deceive machine learning techniques in order to make them produce faulty or erroneous outputs. Adversarial attacks exploit weaknesses in the mathematics behind learning algorithms, and not weaknesses in the operating system or application program itself. This is because of the fact that knowledge is based on the statistical patterns and not any kind of rule-based reasoning, and thus input or datasets can manipulate the model greatly. The evasion attacks occur during

the inference time when the model is trained and deployed. Here, the attacker introduces perturbations into the input data which lead the model to provide an incorrect prediction. Usually, they are too subtle for a human eye to notice them but significant enough to affect the internal feature representation learned by the model. Some of the examples of how it could be used are changing a road sign in such a way that would be unnoticeable by a human eye but misleading the vision system of the autonomous car that the sign is actually the one showing the speed limit. Such security attacks were proven to be effective against facial recognition, malware detection, medical images classification, and biometric identification systems. Those inputs pass the standard validation process of the program which means that such an attack would go undetected by any regular security mechanisms. An essential takeaway here is that high predictive accuracy does not mean immunity to the adversarial attacks.

13.3.2 Data Privacy and Exfiltration Risks

Machine learning models are typically provided with highly sensitive data, including health information, financial transactions, customer information, company's confidential data, and communications. Typically, the assumption is that by providing only the trained model, it is possible to ensure the privacy of the data that was used to train the model, but the research proved that a lot of information about the original dataset can be easily recovered from the trained model. But these privacy issues became one of the major barriers for modern AI.

A. Model Inversion and Membership Inference Attacks

Model inversion attacks attempt to "invert" the attributes of the data on which the machine learning model is trained based on the output from the model. Iterative interaction with the model enables an attacker to gain access to the features of the data or the approximation of the original training sample. For example, it was shown that it is possible to reconstruct a face using a model that only outputs confidence scores in facial recognition systems. However, the privacy problem that is addressed with membership inference attacks is quite different. Rather than trying to reconstruct the training data, the attacker attempts to determine whether or not data for a particular individual were included in the training dataset. Membership inference attack takes advantage of subtle differences in the response of the model to the samples that it has seen before and to the ones that it hasn't. Knowing whether someone belongs to the training set could potentially be extremely private, especially when the dataset is a medical record, a financial record, or some other personal attribute.

B. Model Extraction and Intellectual Property Theft

The development of machine learning models of good quality is a costly process in terms of the data gathering, computing resources, and professional involvement. Therefore, trained models are intellectual property assets. Model extraction attacks are the ones which seek to replicate the proprietary model through iterations via the many calls to a publicly available prediction service. A hacker studies the output generated on the basis of a small number of inputs and gradually replicates the behavior of the proprietary model, developing another but equally functional model. The replicated model can then be freely used, ignoring the licensing policies which apply to it, and permitting the competitors to use the proprietary technology without spending the money required by the original developer. Besides being financially damaging,

model extraction opens up the attack vector since it provides a possibility to analyze the model offline and find its weaknesses in order to conduct further attacks.

C. Governance and Operational Vulnerabilities

In addition to technical problems, there are governance and operational problems that affect the reliability and accountability of the AI system. Control and transparency of automated systems is key for the security of AI as companies try to entrust important decisions to AI systems.

D. Lack of Explainability

AI systems, particularly the deep neural networks, are sophisticated statistical models that make many decisions which are difficult to comprehend. Such models have the ability to generate highly accurate predictions, but they do not provide an explanation of the predictions. This is known as the black box problem. The lack of an explanation presents some significant operational and regulatory issues. There is no way of knowing how a certain decision was generated, thus incorrect, unfair or manipulated decisions would go unnoticed. In industries such as the medical industry, financial industry, and criminal justice, human intervention in the process of decision-making is one of the most important elements. In pursuit of developing explainable AI systems, interpretable models, feature attribution methods, and human intervention play significant roles.

13.4. COMPOUNDED VULNERABILITIES AT THE INTERSECTION OF BLOCK CHAIN AND AI

13.4.1 AI-Driven Exploitation of Block chains

A. Automated Smart Contract Vulnerability Detection and Exploit Generation

Traditionally, exploits for smart contracts were found by specialized human auditors spending entire days manually auditing either Solidity or Vyper code. Nonetheless, recent advances in machine learning have greatly altered the threat landscape in question. Attackers are actively training their personalized GPTs and GNNs on large data sets of previously exploited vulnerabilities, reentrancy attacks on smart contracts, and known compiler bugs. These AI-powered models analyze new bytecodes of the newly-deployed smart contracts in real-time via public block explorers (e.g. Etherscan).

Once a smart contract is deployed, the AI model of an attacker analyzes it's CFG and AST tree looking for logic vulnerabilities, like unhedged reentrancy vulnerability or invalidated state variables. As soon as an AI model finds some problem with a contract, it doesn't simply notify an attacker about that. It uses reinforcement learning algorithms in order to automatically produce an exploit for that vulnerability. As a result, zero-day exploits may appear almost instantly after deploying the target smart contract.

B. High-Frequency Market Manipulation and Arbitrage Extraction in DeFi

The DeFi platforms leverage open, transparent liquidity pools based on AMM formulae. Such structure makes DeFi platforms particularly susceptible to AI-based attacks. There is now malicious infrastructure that leverages deep reinforcement learning techniques to observe the mempools where unconfirmed transactions are stored before being executed. These AI-powered systems are able to precisely forecast the effect that a particular user transaction will have on the price of assets on several DEXs. With the use of high-frequency trading bots, a trader is able to launch predatory front-running and "sandwich attacks." In case an AI system identifies

a big buy transaction for a particular asset, it initiates its own buy transaction with higher gas fee, so that a validator prioritizes it and increases the price for the original user.
As soon as the original user transaction is completed at the increased price, the bot sells its transaction and gains a specific amount of profit. Such Miner Extractable Value syphoning makes millions from honest retail users every day.

13.4.2 Block chain Failures Impacting AI Systems

A. Decentralized Oracle Manipulation and Data Poisoning

AI models operating in a decentralized manner cannot connect to the outside world; for this purpose, AI models need decentralized oracles (such as Chainlink), which give them data from the outside world – be it the prices of stocks, weather information, or records from Internet of Things sensors. In case an attacker is able to hack the oracles, it undermines the whole security system of the AI connected system.
The moment a smart contract is deployed, the attacker's AI model scans its CFG and AST and discovers any potential vulnerabilities that might be present in the code, including any unconsidered vectors of re-entrancy or invalidated state variables.
After finding a vulnerability, the model does not stop here but applies reinforcement learning techniques to craft a payload for the attack. That way, zero-day exploits can be developed and used right after deploying a smart contract.

B. High-Frequency Market Manipulation and Arbitrage Extraction in DeFi

The DeFi platforms depend on the public liquidity pools controlled by AMM formulas. This makes them very exposed to AI exploitation. Nowadays, the malicious trading systems use deep reinforcement learning algorithms which scan public mempools for any transactions that await validation. The AI algorithms allow predicting how a pending transaction would move asset prices on different DEXs. With the help of automated HFT bots, it is possible to conduct front running attacks and "sandwiches". The moment when a large amount of buy transactions for some asset is detected by the AI algorithm, a transaction with higher fees is made. Thus, the validator prioritizes the malicious transaction and inflates the asset price.
Then, the moment after the user's transaction is executed at the inflated price, the AI bot executes sell orders and makes a precise profit. Thus, the mining extractable value (MEV) is extracted by hackers and makes billions of dollars' worth of money from honest retail users.

13.4.3 Block chain Failures Impacting AI Systems

A. Decentralized Oracle Manipulation and Data Poisoning

An artificial intelligence system that is deployed in the decentralized world can never communicate with the external world on its own; therefore, such an artificial intelligence will have to use decentralized oracles (for instance, Chainlink) for providing information about the real world, like stock prices, weather, IoT sensors logs and more. Should an oracle get hacked by an attacker, then the entire AI network will become vulnerable to attack. After deploying the smart contract, the artificial intelligence model of the attacker maps the CFG (Control Flow Graph) and AST (Abstract Syntax Tree) of the smart contract in order to find out any logical weaknesses in the smart contract. These logical vulnerabilities may include anything like neglected re-entrancy exploits or invalidation of state variables. After identifying any such

weakness, the AI will not merely report about the vulnerability but also generate a payload for exploiting the vulnerability through reinforcement learning.

B. High-Frequency Market Manipulation and Arbitrage Extraction in DeFi

In terms of DeFi protocols, their liquidity pools are publicly visible and controlled by Automated Market Maker formulas. This type of system leaves them exposed to the risk of manipulation by AI systems. Today, malicious trading infrastructure leverages deep reinforcement learning algorithms to analyze publicly available memory pools (mempools). The pending transactions in this pool allow the machine learning model to calculate with great accuracy how a particular user transaction will affect the asset prices at several DEXs. With the help of automated high-frequency trading robots, it becomes possible to execute such attacks as predatory front-running and "sandwich attacks." When a model detects a large-volume buying transaction for a certain asset, it automatically places its buy transaction, but with a higher gas fee, which makes validators prioritize its transaction. As a result, the price of the asset increases for the original user. Right after executing the original user's order, the bot sells its share and earns an exact amount of money. The Miner Extractable Value extraction process takes millions of dollars out of the hands of honest retail users each day.

13.4.3 Block chain Failures Impacting AI Systems

A. Decentralized Oracle Manipulation and Data Poisoning

AI-based systems which have been implemented in decentralized settings do not have direct interaction capabilities with the outside world. In order to get some real-world data like the prices of the stocks, weather forecast, and IoT sensor logs, the decentralized oracles (Chainlink is one of them) are used. In case an attacker compromises these oracles, then the security of the whole connected AI system becomes compromised. For example, in case of using a supply chain platform powered by an AI algorithm, the attacker may launch a flash loan attack which allows him/her to manipulate the price metrics or inventory logs of the decentralised oracle network. As a result, this compromised data is written by the oracle straight to the block chain ledger.

In case if the connected AI system uses the ledger data in order to constantly train itself and become fine-tuned, this single corrupt entry becomes a source of bias. It is a sophisticated data poisoning technique which involves manipulation of the decentralized consensus layer used by the AI system.

B. Decentralized Storage Latency and Distributed Inference Degradation

To ensure the security of centralized cloud storage, developers tend to utilize decentralized storage networks, such as IPFS, File coin, or Arweave, where huge AI models and massive training datasets are stored. Though the decentralization allows cryptographic hashing to ensure immutability of the data, this approach entails additional latency when retrieving the information. If the AI model processes complex inference procedures at the edge of the network, for example, the process of detection of objects by autonomous delivery drones or detecting fraudulent transactions in high-frequency flows of transactions, it requires retrieving the model weights and metadata shards from different nodes of the distributed system all around the world. Due to irregular spikes of latency while synchronizing the nodes of decentralized networks, the data pipeline encounters micro-delays. In case of time-sensitive AI models, the delay may result in interruption of the inference process, making the system slow and non-responsive to crucial events.

13.4.3 Smart Contract-Driven AI Risks

A. Autonomous Malicious AI Agents and Out-of-Control DAOs

Once an AI model has been embedded within a block chain smart contract, then the AI model is able to get its own crypto currency public key and working digital wallet. The end result is that the AI model is able to become an independent economic entity. Although there are many positive implications of this, there are also some severe security concerns. Once an autonomous AI entity within a DAO gets the objective function corrupted, then it may use its funds to inflict actual damage on the real world without any human intervention.

13.5. ADVANCED COUNTERMEASURES AND MITIGATION FRAMEWORKS

13.5.1 Securing the Block chain Layer

Protecting the block chain layer is crucial in providing dependable and trustworthy decentralized systems. Although block chain technology ensures transparency and immutability of data, weaknesses in smart contracts and consensus models can put blockchain networks in danger of being exposed to numerous security risks. For this reason, secure software development and robust consensus algorithms are key components of successful protection. Formal verification is one of the most efficient tools, which uses mathematically rigorous techniques for guaranteeing that the smart contract works in a certain way for all possible execution environments. In contrast to the traditional software testing approach, formal verification allows for identifying logical inconsistencies, re-entrancy problems, and other arithmetic errors at an early stage. It is also vital to implement resilient consensus algorithms. Proof of Stake is more energy-efficient compared to the Proof of Work algorithm, as validators are chosen according to the amount of their staked assets, and economic sanctions deter malicious activity. Proof of Authority is popular among private and consortium blockchains, providing efficient transaction validation with high resilience to identity-based attacks.

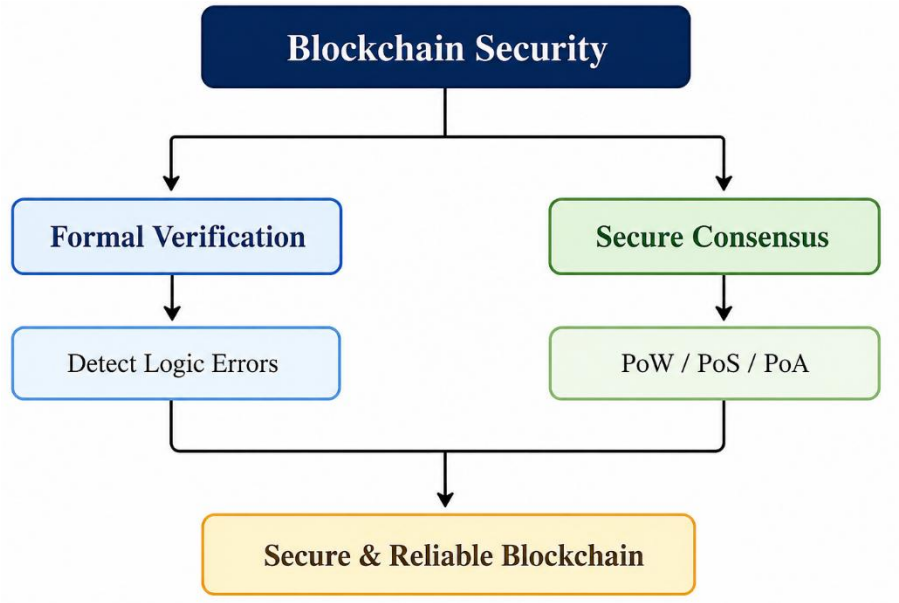


Figure 1. Blockchain security layer Architecture

Moreover, frequent security audits, private key management, and continuous network surveillance also play an important role in enhancing the security of block chain technology. In this way, a multi-layered security framework is created for ensuring the resilience, scalability, and enhanced security of block chain systems.

13.5.2 Securing the AI Layer

Protection of AI is a layered approach of defense in depth which entails the protection of the whole lifecycle of machine learning. It is not enough to rely on one type of security since the threats could target the training data, the parameters of the model, the method of inference or even the prediction service.

A. Adversarial Training and Robust Optimization

Adversarial training is probably one of the most effective strategies for improving the robustness of a model. As part of the model design process, adversarial samples are intentionally included into the training set which provides an opportunity for the algorithm to be adapted to malicious manipulations. The model is trained in a repetitive way using both beneficial and malicious samples, so that it will gain resistance to the attacks and increased robustness in the adversarial setting. Additional strategies of robust optimization provide protection against attacks in the form of ensuring the insensitivity of the prediction to any small variations of the input data. All these regularization techniques, certified robustness, defensive distillation, and ensemble learning improve robustness of the model under adversarial attacks without degrading the performance in the standard setting. Adversarial training does not prevent all attacks, but it significantly complicates their realization for the attacker while increasing their computational cost.

B. Privacy-Preserving Machine Learning

The protection of sensitive training data is equally important. The machine learning approaches are privacy-preserving in nature, thus making collaborative modeling possible while minimizing the use of confidential information. In the case of Differential Privacy (DP), proper amounts of statistical noise are added to the learning or querying process to ensure that each individual datum's sensitivity to the query or model output is just a few proportions. Mathematically proven privacy guarantees are achieved without degrading the utility of the trained model. Federated Learning (FL) partitions the training process through training being carried out locally by devices or organizations on their data and the sending and aggregation of only model updates or gradients on the centralized server, rather than sending raw data. Not only does it mitigate privacy concerns, but it also makes it possible to learn geographically distributed entities collaboratively. Homomorphic Encryption (HE) allows for operations to be performed on encrypted data without decrypting it. As a result, cloud-based AI services can carry out inference or training processes using only encrypted data. Although very resource-intensive, advances in the homomorphic encryption technology make it viable for use in privacy-sensitive areas like health care, finance, and governmental systems.

13.5.3 Unified Defence Frameworks

In order to provide for privacy, transparency, and accountability while keeping the information secret, certain security mechanisms will be required to integrate AI and block chain. One possible way to do so is implementing Zero-Knowledge Proofs (ZKPs) for AI-powered block

chains. The goal of the zero-knowledge proofs is for one party (prover) to provide the other party (verifier) with proof of the computation without revealing any data, model parameters, or intermediate computation steps. It is particularly important to implement ZKP in the case of AI when the machine learning model may operate using some highly sensitive information such as medical, financial, industrial data, or personally identifiable information. Instead of disclosing the whole dataset or proprietary neural network architecture, ZKP makes it possible for AI developers to provide verifiable cryptographic proof of their model training, inference, and parameter updates being performed in compliance with some protocol. Each stage of the AI process, starting from dataset registration, through pre-processing, feature engineering, training, validation, deployment, and inference can be backed up with the verifiable cryptographic proof stored in the block chain forever.

The recent breakthroughs with zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge), Recursive ZKPs, and zkML (Verifiable Machine Learning) have enabled to significantly reduce the overhead required to prove something, increasing the feasibility of privacy-preserving AI verification. The technologies help to build decentralised AI marketplaces, federated learning setups, autonomous agents, and inter-organisation AI collaborations while ensuring verifiability, explainability, and privacy-preserving AI decisions. In the future, the ZKPs will become crucial for secure AI governance and regulation, decentralised model monitoring and inspection, and confidential machine learning and data processing in Web3. In the future, the ZKPs will become crucial for Web3, helping to govern and regulate AI securely, monitor and inspect AI models, and process data confidentially.

13.6. FUTURE TRENDS AND OPEN RESEARCH DIRECTIONS

With the increasing integration of block chain and AI technology, the future of secure digital systems is evolving quickly. The research challenges keep evolving with evolving technology, cyber threats and regulatory needs. In order to create next generation intelligent infrastructures, the future security framework will have to incorporate the following elements: Cryptographic robustness, Decentralized governance, Trustworthy AI and Privacy-preserving computing.

13.6.1 Post-Quantum Cryptography (PQC)

Quantum Computing is among the most disruptive technologies of the contemporary cyber security era. If Shor's algorithm were to be effectively implemented through large-scale quantum computing, ECDSA, ECC, and RSA, for example, would become weak. In addition, Grover's algorithm might expose cryptographic hash functions to brute force attacks since it would increase efficiency. With regard to block chain systems, quantum attacks might affect their authenticity, identity management, smart contract security, and even decentralized consensus mechanism. Without post-quantum solutions being applied, quantum attackers would be able to forge signatures and compromise wallets. Similarly, AI technologies are also vulnerable. If an opposing party manages to penetrate the quantum level, it can crack private training pipelines, confidential communication of distributed federated learning, model parameters, and encrypted AI data. Given the increasing deployment of AI in various industries, such as medicine, finance, defense, and smart cities, quantum-safe security becomes an

essential issue. In order to prevent such breaches, scientists create algorithms of Post-Quantum Cryptography (PQC), which rely on the computational problems that can be considered robust to quantum computing. These include such algorithms of PQC as: multivariate cryptography, code-based cryptography, isogeny-based cryptography, hash-based signatures, and lattice-based cryptography. In particular, CRYSTALS-Khyber (key encapsulation) and CRYSTALS-Dilithium (digital signatures) are the examples of standards that are emerging under the auspices of the NIST Post-Quantum Cryptography Standardization Project. Some of the open research issues include the development of quantum-safe blockchain protocols, quantum-safe smart contracts, quantum-safe AI communication, and hybrid classical-quantum security architectures.

13.6.2 Regulatory and Standardization Standards

Good governance is critical in ensuring responsible use of the block chain and AI. Various governments, international bodies, and standards have made significant efforts towards developing policies for transparency, accountability, privacy, fairness, and cyber security in intelligent and decentralized systems. According to the European Union AI Act, there exists a regulatory framework that establishes a risk-based approach for AI that is trustworthy and gives the requirements of transparency, human oversight, robustness, and risk management based on the effects of the applications on society. On the other hand, the NIST AI Risk Management Framework (AI RMF 1.0) provides ways of creating safe, explainable, and accountable AI throughout their life cycle. Some of the governance standards include the IEEE 7000 Series, ISO/IEC 42001 (Artificial Intelligence Management Systems), ISO/IEC 23894 (AI Risk Management), and ISO/IEC 27001 (Information Security Management). The World Wide Web Consortium (Verifiable Credentials), World Wide Web Consortium (Decentralized Identifier (DID)), ISO/TC 307, and the European Block chain Services Infrastructure (EBSI) are making strides towards interoperability, decentralized digital identity, security of data sharing, and trustable cross-border services in the field of block chain technologies. The investigation of AI governance in relation to DAOs, regulatory interoperability across borders, privacy preserving techniques for compliance verification based on Zero-Knowledge Proofs, auditing AI decisions, responsible Generative AI, governance of autonomous AI agents, and digital trust infrastructures using Block chain technology will remain a focus area in the future.

13.7. SUMMARY OF KEY INSIGHTS

The coming together of the technologies of block chain and artificial intelligence represents a breakthrough in the design of future-generation computer systems. Nevertheless, as discussed in this chapter, the merging of the two very strong technologies leads to a very intricate maze of security concerns. These vulnerabilities are not only additive but multiplicative in nature since they affect the fundamental logics of both systems. Blockchain systems are still structurally susceptible to exploits at the protocol level, consensus attacks (51%, selfish mining), and smart contract bugs (re-entrance vector). These vulnerabilities have become targets for automated scanning by AI tools which can discover and exploit weaknesses within seconds. Artificial intelligence systems struggle with the "black box" problem, leaving them vulnerable to adversarial machine learning attacks, data poisoning, and model exfiltration. These vulnerabilities become much more dangerous when locked into an immutable ledger.

The intersection layer creates severe compounded risks. Oracle manipulation can poison AI training sets via decentralised consensus, while autonomous AI agents equipped with crypto-wallets can launch self-funded, unstoppable cyberattacks if their objective models are corrupted. Advanced mitigation frameworks require a layered, multidisciplinary defence strategy. Securing these hybrid systems requires combining formal smart contract verification, adversarial training, and privacy-preserving tools like Differential Privacy, federated learning, and Homomorphic encryption. Finally, Zero-Knowledge Proofs (ZKPs) are vital for establishing a private, verifiable audit trail for AI computations directly on the block chain ledger. As we look toward the future of decentralized computing, the convergence of block chain and AI will continue to accelerate, driven by the emergence of autonomous Web3 economies, decentralized machine learning networks, and edge computing architectures. However, the long-term success of this digital evolution depends entirely on a proactive approach to security hygiene. Engineers, researchers, and enterprise architects must abandon the old habit of treating security as a patchwork fix applied late in the development cycle. Instead, they must design these hybrid systems with security in mind from day one.

This requires implementing strict verification pipelines for training data, building automated fail-safes into autonomous smart contracts, and continuously testing models against adversarial attacks. Furthermore, as quantum computing advances, the cryptographic baselines protecting both block chains and AI models will face new threats. This transition will require a rapid shift toward lattice-based post-quantum cryptography (PQC). Ultimately, the goal is not to slow down innovation but to build a resilient, self-defending digital infrastructure. By anchoring probabilistic AI models within the secure, deterministic boundaries of advanced cryptography, the global technology community can build autonomous networks that are both highly innovative and exceptionally secure.

CHAPTER 14

PRIVACY-PRESERVING MECHANISMS IN DECENTRALIZED INTELLIGENCE

Contributed by: Samrat Sahana, Sukanya Basu, Uddipta Modak

14.1. INTRODUCTION

The paradigm of distributed computation is undergoing a profound shift. We are rapidly moving away from architectures controlled by a single centralized authority toward intelligent, autonomous systems powered by Artificial Intelligence (AI) and blockchain technology. This transformation is driven primarily by the natural convergence of these two fields. Decentralized intelligence frameworks offer significant advantages by eliminating single points of failure and dismantling data monopolies. In these environments, the failure of an individual node does not compromise the entire network, and no single entity retains exclusive control over the underlying data assets. However, these distributed architectures introduce novel security vulnerabilities and remain susceptible to sophisticated adversarial exploits. For instance, malicious actors frequently target these networks using gradient leakage, membership inference, label leakage, and model inversion attacks to extract confidential information. This chapter systematically examines the methodologies required to maintain privacy and security within decentralized learning environments. We explore how smart contracts can eliminate centralized parameter servers, establishing a secure and verifiable framework where AI and blockchain technology seamlessly co-exist. Furthermore, we discuss critical cryptographic primitives—such as Fully Homomorphic Encryption (FHE) and Secure Multi-Party Computation (SMPC)—alongside algorithmic solutions like Differential Privacy (DP), while addressing the high computational and network overheads inherent to these approaches. We evaluate the deployment of Zero-Knowledge Proofs (ZKPs) to enable trustless computation without exposing private source data, and we examine how Non-Interactive Zero-Knowledge Arguments of Knowledge (NIZKPs) can reduce transaction footprints and lower gas costs. Finally, the chapter investigates hardware-level isolation mechanisms enforced by internal silicon controllers and outlines algorithmic defenses to mitigate execution risks. Under the proposed Privacy-Preserving Decentralized Primal-Dual algorithm with Positive-Incentive Noise (PP-DPIN) framework, node privacy configurations are dynamically tailored based on network topology. This approach regulates the volume of training data transmitted and limits total information leakage via a rigorous privacy budget accounting mechanism. Our analysis maps the critical trade-offs between computational expenditure, communication overhead, and

model accuracy, offering a blueprint for selecting optimal architectures for secure decentralized machine learning.

The integration of Artificial Intelligence (AI) and blockchain technology has emerged as a cornerstone of recent research in distributed systems. As a foundational layer, a blockchain provides a decentralized, immutable ledger. Concurrently, AI frameworks offer the capacity to detect intricate patterns, generate predictions, and automate reasoning across expansive datasets. Combining these methodologies yields intelligent architectures where autonomous agents learn, reason, and coordinate within a trust-minimised environment. This paradigm addresses critical challenges surrounding data integrity, transparency, and structural accountability. By utilizing smart contracts, systems can enforce strict accountability protocols among independent agents and audit transaction executions across the network. Consequently, decentralized blockchains enable comprehensive auditing of decisions made by autonomous AI models, ensuring that algorithmic execution remains transparent and trustworthy. The global volume of data is expanding at an unprecedented rate, driven incessantly by the proliferation of Internet of Things (IoT) devices. Historically, machine learning models were trained by aggregating diverse data streams into centralized cloud data lakes. While computationally convenient, this practice introduces severe security liabilities, including data breaches, unauthorized leakages, and regulatory compliance failures. Decentralized intelligence addresses these vulnerabilities by shifting toward alternative training topologies, most notably Blockchain-powered Federated Learning (FL). Federated Learning natively aligns with modern data compliance standards while continually refining the performance of a shared global model. Instead of transmitting raw datasets, FL-enabled devices process data locally and compute on-device gradients. The local updates and model parameters are then shared across the decentralized network, where smart contracts manage the aggregation process. This structural shift eliminates the single point of failure typically associated with centralized parameter servers.

This architectural innovation can be effectively leveraged across several domains:

- **Healthcare Systems:** Remote hospitals can collaboratively develop diagnostic models via blockchain-backed federated networks without exposing confidential patient records.
- **Financial Infrastructures:** Smart contracts can automatically execute financial transactions by encoding complex risk models derived directly from machine learning.
- **Industrial IoT:** Industrial machinery can record sensor-based inferences onto an immutable ledger to generate verifiable audit trails for critical operations.

The inherent decentralization of these consortia effectively prevents any single participant from manipulating outcomes or deleting historical records. Despite these benefits, public blockchain architectures introduce a fundamental structural challenge to data privacy. The requirement of a public ledger for independent verification directly contradicts the principle of data minimization, as verification data cannot be removed from an immutable audit trail. These privacy risks intensify when such ledgers manage data pipelines for intelligent networks. If on-chain smart contracts are subjected to adversarial probing, machine learning models may

memorize and inadvertently leak training attributes. Furthermore, coordination vectors and smart contract state variables remain directly accessible on-chain, exposing the system to competitive analysis and target re-identification. This landscape is further complicated by stringent international data protection mandates. Legal requirements such as the explicit right to be forgotten, the right to rectify data, and data processing restrictions—as codified under the General Data Protection Regulation (GDPR)—stand in direct structural conflict with blockchain immutability. Therefore, privacy cannot be treated as an optional feature added in the later stages of deployment; it must be an inherent property of the system's architecture.

14.2 FOUNDATIONS OF DECENTRALIZED INTELLIGENCE

Decentralized intelligence is formally defined as the convergence of distributed computing, artificial intelligence, and decentralized ledgers. This paradigm entails deploying machine learning algorithms across environments where state synchronization and operational trust are maintained via a distributed consensus algorithm rather than a centralized entity. Decentralized intelligence fundamentally diverges from traditional cloud-hosted AI through three core structural pillars:

1. The distribution of machine learning algorithms across independent nodes.
2. The reliance on a decentralized consensus mechanism for coordination.
3. The execution of algorithmic operations directly over encrypted data fields.

These pillars are sustained by the following operational characteristics:

- **Data Locality:** Source data remains local to its point of origin and under the exclusive custody of the generating entity, structurally prohibiting unauthorized data aggregation.
- **Verifiable Coordination:** The logical sequence of model training, optimization, and inference is recorded openly and remains completely auditable via distributed consensus.
- **Multi-Stakeholder Governance:** The global model, training algorithms, and underlying parameters are collectively governed by a distributed network of stakeholders.

During optimization, the architecture refines the global loss function without requiring nodes to disclose their private training sets. Smart contracts replace centralized aggregation servers with a network of validating nodes, rendering the aggregation process entirely transparent and traceable. Because data on a public blockchain is openly accessible, the system must deploy robust cryptographic primitives to shield the parameters broadcast across the network. Additionally, consensus validation rules guarantee that no model update can bypass verification, effectively excluding malicious or malformed parameters from the global state.

14.2.1 Architectural Integration Patterns

The integration topologies combining blockchain and machine learning are broadly classified into three structural patterns:

- **On-Chain Coordination:** The blockchain serves strictly as a coordination layer and a cryptographic audit trail. It stores cryptographic hashes of model updates, node

metadata, and aggregation schedules on-chain, while the resource-intensive training and inference workloads are executed entirely off-chain.

- **On-Chain Inference:** Smart contracts natively evaluate compact machine learning models or execute the results of more sophisticated architectures. While this approach minimizes trust requirements, its capabilities are tightly restricted by the computational limits of the blockchain's consensus mechanism, restricting practical use cases to low-dimensional neural networks or basic decision trees.
- **Data Marketplaces:** The blockchain functions as a settlement layer for data transactions. This pattern tokenizes and commoditizes access to data assets and model inferences without storing the underlying raw assets on the shared ledger.

14.2.2 Stakeholders and Data Flows

Decentralized intelligent frameworks encompass four primary stakeholder categories:

- **Data Owners:** The entities that maintain exclusive custody and control over the raw source data.
- **Model Trainers:** Participants who dedicate computational resources to fine-tune and optimize model weights.
- **Ledger Holders:** The network validators responsible for running the consensus protocol and maintaining the state.
- **Model Consumers:** The end-users or systems that query the framework to generate inference results.

Notably, privacy leakages can manifest at any stage of the operational pipeline where data becomes accessible. Vulnerabilities are particularly acute during initial data collection, parameter interaction, transaction execution, and the consumption of final inference results.

14.2.3 Hardware-Enforced Enclaves and System Topology

To bridge the operational gap between off-chain data generation and on-chain coordination, modern solutions deploy a multi-tiered physical topology. At the foundational layer, data extraction and initial processing occur within hardware-isolated execution environments (TEEs), such as Intel SGX or AMD SEV. By design, memory encryption keys within these enclaves are inaccessible to the host system, preventing local platform operators from directly extracting model states. Instead of exposing raw parameters, the TEE generates an isolated cryptographic attestation proof of its execution, which is transmitted to the cryptographic aggregation layer for ultimate settlement on the blockchain consensus layer. This settlement procedure maintains an unbroken chain of trust across the architecture. Finally, lower-level silicon controllers reinforce memory isolation, ensuring that no outer operating system layer can inspect or manipulate the contents of the secure execution environment.

14.3 THREAT MODELS IN DISTRIBUTED AI

Designing resilient privacy mechanisms requires a comprehensive taxonomy of the specific threat landscape confronting decentralized learning frameworks.

14.3.1 On-Chain Data Exposure

A primary vulnerability in public blockchain networks stems from the permanent exposure of sensitive data or linkable cryptographic identifiers uploaded to public ledgers. Transactions are propagated across an open network of validating nodes and recorded in perpetuity, preventing subsequent deletion. This vulnerability is compounded by the fact that pseudonymous addresses remain susceptible to de-anonymization via third-party graph analysis and metadata correlation.

14.3.2 Gradient Leakage and Model Inversion

Transmitting locally computed gradients instead of raw datasets drastically reduces communication overhead, yet intermediate parameter updates remain highly vulnerable during transit. Gradients encode the precise optimization paths taken during training, which adversaries can exploit to execute model inversion, membership inference, and property inference attacks. Using iterative optimization algorithms, an attacker can leverage publicly available parameter updates to reconstruct training inputs. For instance, the Deep Leakage from Gradients (DLG) method iteratively synthesizes dummy gradients to approximate the publicly broadcast gradient vectors. DLG has demonstrated the capacity to reconstruct private training images and text sequences. Within a federated network, a malicious actor can exploit these parameter updates to reconstitute private samples or determine whether a specific record was included in a node's training set. Additionally, labels can be leaked directly from the signs of the gradients in the final classification layer, allowing an adversary to infer the presence of specific target classes within a training batch.

14.3.3 Membership Inference Attacks (MIA)

Membership Inference Attacks (MIAs) enable an adversary to determine whether a specific user's record was used to train a local model node. These exploits leverage the characteristic statistical variances in an optimized model's confidence scores, which typically differ for inputs included in the training set versus unseen data. Because user anonymity across the network depends on the indistinguishability of individual contributions, an attacker can analyze these variances to identify data contributors, thereby compromising private user profiles.

14.3.4 Identity Linkage and De-Anonymization

Decentralized networks must manage user identities to distribute financial rewards, weight gradient updates, and enforce access controls. However, if the architecture relies on persistent public keys or wallet addresses as identity proxies, adversaries can correlate transaction histories across multiple decentralized applications. In small consortium networks, metadata correlation can rapidly compromise identity anonymity, mirroring the structural re-identification risks characterized by classical k-anonymity models.

14.3.5 Transparency–Confidentiality Conflicts in Smart Contracts

Smart contracts execute deterministically, and their state transitions are made entirely visible to facilitate independent validation across the network. This open access introduces an architectural contradiction when the contract logic depends on confidential inputs, such as proprietary risk models, financial bids, or private medical variables. In the absence of privacy-

preserving frameworks, any network node can replay contract executions to extract these private inputs, undermining the confidentiality of the data layer.

14.3.6 Regulatory and Compliance Tensions

The legal right to modify or delete data under modern data protection frameworks is structurally impeded by blockchain immutability. To achieve regulatory compliance, decentralized intelligent architectures must implement specific design configurations. These include utilizing non-revocable cryptographic commitments on-chain while storing private data in off-chain repositories governed by revocable confidentiality parameters. Furthermore, the permanent record-keeping nature of distributed ledgers creates barriers to the right to rectification, as data incorrectly committed to the ledger state cannot be altered or erased out-of-band.

14.3.7 Adversarial Profiles and Collusion Boundaries

To mitigate these vectors, threat frameworks classify adversaries based on their behavioral capabilities. A fundamental distinction is made between Honest-but-Curious (semi-honest) actors and Malicious (Byzantine) nodes. Semi-honest adversaries adhere strictly to protocol rules but attempt to mine sensitive data through passive observation. Conversely, Byzantine nodes actively inject toxic parameters, malformed gradients, or sybil identities to hijack the global state. Furthermore, decentralized architectures rely on strict collusion boundaries to maintain trust. Cryptographic threshold schemes are mathematically configured to ensure that colluding malicious nodes do not exceed a definitive ratio of the total validator network, preventing unauthorized data reconstruction.

14.4 CRYPTOGRAPHIC PRIVACY-PRESERVING PRIMITIVES

Cryptographic primitives isolate local updates, securing them during transit and throughout decentralized aggregation phases.

14.4.1 Fully Homomorphic Encryption (FHE)

Fully Homomorphic Encryption (FHE) permits arbitrary mathematical operations to be performed directly on encrypted ciphertexts. Within a blockchain-anchored intelligence framework, edge nodes encrypt their local model updates with a network public key before broadcasting them. Smart contracts or distributed aggregators then compute mathematical averages directly over these ciphertexts. The aggregated global model is only decrypted collectively once the network satisfies its consensus threshold rules. While FHE provides rigorous security against passive eavesdropping, its high-degree polynomial arithmetic introduces substantial computational overhead. This burden challenges resource-constrained edge hardware. Furthermore, the structural expansion of ciphertexts increases network transmission stress, which can cause severe latency bottlenecks across distributed validation clusters.

14.4.2 Secure Multi-Party Computation (SMPC)

Secure Multi-Party Computation (SMPC) enables a group of mutually distrustful parties to jointly evaluate a target function over their private inputs without revealing the underlying data.

Utilizing frameworks like Shamir's Secret Sharing, a local gradient vector can be divided into multiple polynomial shares and distributed across the network. Reconstructing the true gradient requires the explicit cooperation of a pre-configured threshold of honest nodes. In decentralized AI systems, SMPC shields the federated aggregation phase, ensuring that the global update is only visible in its aggregate form. However, this approach demands significant peer-to-peer communication bandwidth, resulting in high operational latency.

14.4.3 Zero-Knowledge Proofs (ZKPs)

Zero-Knowledge Proofs (ZKPs) enable a prover to mathematically demonstrate the validity of a statement to a verifier without disclosing the underlying data variables. By deploying non-interactive zero-knowledge proofs (NIZKPs), these arguments can be integrated directly into smart contract transactions. This allows a node to prove to the contract state that its gradient computations and inferences were executed accurately within specified parameters. Consequently, this methodology resolves challenges surrounding data poisoning and Sybil attacks without revealing private training attributes. Furthermore, NIZKPs compress transaction footprints, allowing smart contracts to optimize the gas expended on proof validation.

14.4.4 Linear Secret Sharing with Cheating Detection

Standard threshold schemes operate under the assumption of an honest-but-curious threat vector, meaning basic configurations lack mechanisms to audit share integrity during reconstruction. If a malicious node introduces a corrupted share, the resulting global output becomes distorted, leaving the network vulnerable to data poisoning or resource-exhaustion exploits. To mitigate this vulnerability, linear secret sharing schemes integrate cheating detection by pairing polynomial shares with verification variables during setup. These variables serve as implicit commitments to the initial assets. Before computing the aggregate contribution, the orchestrator applies Lagrange interpolation over the validation components. The system then evaluates each share against its initial commitments, discarding invalid inputs before they can alter the global state.

14.4.5 Lightweight Identity-Obscuring Mechanisms

For high-throughput environments where complex cryptographic wrappers are computationally prohibitive, ring signatures and stealth addresses offer a lightweight alternative. These primitives allow a node to sign transactions on behalf of an structurally indistinguishable group. This arrangement conceals the explicit identity of the originator while preserving complete public auditability.

14.5 ALGORITHMIC PRIVACY: DIFFERENTIAL PRIVACY (DP)

Cryptographic mechanisms secure data during active computation; however, sensitive information can still be extracted from a fully trained model via its optimized weights. Differential Privacy (DP) provides a rigorous mathematical framework ensuring that the presence or absence of any single record in a training set causes only a bounded, quantifiable change in the model's output distribution. This methodology injects calibrated statistical noise to restrict the volume of private information an adversary can infer about individual

contributors. Implementing Differentially Private Stochastic Gradient Descent (DP-SGD) introduces a definitive trade-off between privacy and utility. As the magnitude of the injected DP noise increases, the accuracy of the resulting global model may degrade. To analyze how distinct noise components impact the optimization trajectory, developers decompose privacy-enhancing gradient perturbations into two core categories:

- **Positive-Incentive (PI) Noise:** Mechanically generated variables that are cryptographically correlated across the network topology. PI noise is structurally configured to cancel out perfectly during the node aggregation phase. This noise protects local updates against inference attacks without distorting the final convergence path or degrading the accuracy of the global model. Consequently, developers can inject high-variance noise for enhanced localized security.
- **Privacy-Utility Tradeoff (PUT) Noise:** Aggregation-persistent, uncorrelated noise patterns (such as standard Gaussian or Laplacian perturbations). Because PUT noise permanently corrupts the true underlying gradient vectors, it inherently degrades convergence accuracy. Thus, it is used restrictively as a secondary defense layer in vulnerable topologies where perfect noise cancellation is structurally impossible.

14.5.1 The PP-DPIN Algorithm

The Privacy-Preserving Decentralized Primal-Dual algorithm with Positive-Incentive Noise (PP-DPIN) functions as a dynamically adaptive privacy mechanism. Rather than enforcing a uniform policy across the network, it tailors' privacy metrics based on a node's precise position within the system topology.

Internal / Core Node Execution (PI-Driven)

For well-connected core nodes, the framework executes a primal-dual stochastic gradient structure integrated with a Positive-Incentive Noise Generator (PING). During initialization, these nodes leverage additive homomorphic encryption to construct topology-dependent, structurally correlated noise. Throughout local stochastic gradient descent and subsequent peer-to-peer communication cycles, the aggregated noise from neighboring nodes cancels itself out. This allows core nodes to inject high-variance noise to mask local datasets completely without corrupting the global convergence trajectory.

Leaf-Neighbor Node Execution (PUT-Driven)

The algorithm treats leaf-neighbor nodes—nodes that serve as the single point of connection for a degree-one leaf node—differently. Because a leaf node shares only a single communication edge, cryptographic noise cancellation via PING is structurally unfeasible without compromising security. At these interfaces, the PP-DPIN algorithm switches modalities, forcing the injection of independent, uncorrelated PUT noise directly onto messaging lines bound for the leaf nodes. While this introduces a bounded error into the gradient updates, it effectively prevents information leakage across isolated network edges where collaborative masking cannot occur.

Convergence and Performance Characteristics

The operational viability of PP-DPIN is defined by two primary criteria:

- **Convergence Assurances:** The framework provides a formal mathematical guarantee of convergence under both stochastic convex and non-convex optimization objectives.
- **Error Decay & Acceleration:** Within boundary insertions, the leading error terms decay at an optimal rate subject to PUT noise. Furthermore, the system demonstrates a linear speedup relative to the total network size, enabling it to accommodate an expanding number of decentralized nodes without experiencing operational latency spikes.

14.5.2 Formal Privacy-Utility Metrics and Benchmarking

To evaluate these architectures rigorously, engineers track precise empirical metrics throughout each optimization round. Instead of enforcing a static cap, they monitor cumulative privacy expenditure across successive learning iterations using Renyi Differential Privacy accountants, plotting these costs directly against utility metrics like global test accuracy and model convergence timelines. They also dynamically adjust the privacy budget parameter across training epochs. This real-time tracking ensures total information leakage never crosses the system's predetermined maximum threshold. Systems benchmark resistance to gradient attacks by measuring the Structural Similarity Index (SSIM) of adversarial reconstructions. Under Deep Leakage from Gradients (DLG) attacks, a sharp drop in SSIM values provides objective proof that the injected PI and PUT noise configurations successfully protect local datasets. These structural variations mask missing data dimensions entirely, thwarting any reconstruction attempts.

14.5.3 Generalization Techniques

Data generalization techniques—such as k-anonymity, l-diversity, and t-closeness—modify datasets to protect privacy by making any individual record indistinguishable from at least k-1 others regarding quasi-identifiers. While these methods offer less stringent formal bounds than differential privacy, they work well as a lightweight pre-processing step for metadata that requires partial, on-chain disclosure for public auditing or reporting. To counter the structural weaknesses of basic k-anonymity against homogeneity and background knowledge attacks, l-diversity forces equivalence classes to contain at least l well-represented values for each sensitive attribute. Finally, t-closeness prevents skewness attacks by bounding the semantic distance between a group's attribute distribution and the global population distribution.

14.6 TRUSTED EXECUTION ENVIRONMENTS (TEES) & HARDWARE ISOLATION

Trusted Execution Environments (TEEs) secure code and data by providing a hardware-isolated memory enclave that blocks inspection or modification from the host system. This isolation shields sensitive workloads directly from the underlying operating system or hypervisor. Within decentralized intelligent architectures, TEEs allow systems to safely delegate complex tasks—such as evaluating a proprietary model on private input data—to an untrusted node. The node's hardware then generates a verifiable cryptographic attestation to confirm the computation occurred securely inside the enclave. However, microarchitectural side-channel vulnerabilities like Foreshadow, Spectre, and Meltdown still threaten these hardware guarantees. Because malicious host administrators can exploit transient execution side-channels or cache timing analysis to leak enclave memory states, developers must pair

hardware isolation with software-level defenses like constant-time execution paths and compiler-inserted memory barriers.

14.6.1 Architecture and Security Assurances

CPU Configuration Analysis refers to the structural process of assessing, isolating, and validating hardware-level specifications to guarantee execution integrity. Because the physical configuration of the host platform remains vulnerable in hardware-enforced isolation architectures, malicious infrastructure operators can still manipulate the hardware state. To mitigate this, CPU Configuration Analysis evaluates critical hardware components—including core allocations, instruction set architectures, cache topologies, and memory register boundaries—before deploying any workloads. This formal assessment verifies that the physical processor conforms to strict trust profiles. Additionally, operators must implement required firmware microcode updates to patch transient out-of-order execution vulnerabilities, while maintaining isolated cache lines. This rigorous approach ensures that secure computing enclaves process intense data operations safely without risking side-channel attacks or memory leakage.

14.6.2 Intel SGX: Process-Level Isolation Through Enclaves

Intel SGX utilizes a granular, application-centric design that isolates specific user-space processes rather than the broader operating system environment. This approach enables standard applications to establish secure, self-contained execution zones known as enclaves. Because a single physical machine can support numerous independent enclaves simultaneously, the architecture ensures that each enclave remains completely blocked from accessing the memory space of another.

Intel SGX secures this sensitive data by allocating an isolated partition of system RAM called the Processor Reserved Memory (PRM), which houses the Enclave Page Cache (EPC). To safeguard against physical-layer vulnerabilities, a hardware-based Memory Encryption Engine (MEE) automatically encrypts data flowing out to system memory and decrypts it only upon retrieval into the CPU cache. This architecture ensures that physical probing of the memory bus yields nothing but unreadable ciphertext. For integrity verification, a built-in Remote Attestation mechanism embedded directly within the processor's microcode generates a cryptographic hash of the enclave during initialization. External parties can then request a digitally signed verification payload, or "Quote," which uses a global Public Key Infrastructure (PKI) to confirm that authentic code is executing securely on genuine hardware.

14.6.3 ARM TrustZone: Dual-World System Partitioning

ARM TrustZone employs a macro-level, system-wide approach to hardware separation by duplicating physical hardware states rather than confining individual application processes. The architecture splits the platform into two distinct execution environments:

- **The Normal World (Non-Secure):** A partition dedicated to managing standard, feature-rich operating systems—such as Linux, Android, or iOS—alongside everyday user applications.

- **The Secure World:** An isolated zone reserved for running a compact, specialized Trusted OS, like sepOS or OP-TEE, and its corresponding secure software applications.

Unlike systems that rely on continuous cryptographic RAM encryption, ARM TrustZone shields sensitive operations using strict, physical access controls. By leveraging the architecture's memory management unit (MMU) and memory controller, it prevents the standard operating environment from ever addressing, mapping, or interacting with secure memory segments. Furthermore, TrustZone doesn't rely on a uniform, silicon-level remote attestation standard. Instead, the framework prioritizes the clean separation of execution environments. Validation and security stem from a platform-specific Secure Boot sequence, which device manufacturers manage independently through their own proprietary cryptographic keys. This setup guarantees that edge intelligence nodes maintain strict environment integrity right up until a sync event occurs.

14.7 IDENTITY AND ACCESS CONTROL LAYER

14.7.1 Decentralized Identity Mechanisms

Decentralized Identifiers (DIDs) and Self-Sovereign Identity (SSI) frameworks eliminate the need for centralized identity registries by letting participants manage their own cryptographically verifiable identifiers. Backed by the formal W3C Decentralized Identifiers specification, these mechanisms let network participants interact safely without putting their real-world identities at risk. Furthermore, layering zero-knowledge proofs onto verifiable credentials enables selective disclosure. A node can instantly verify critical attributes—like whether a participant is authorized or compliant—without ever seeing the private identifying data behind them.

14.7.2 Access Control and Consent Management

Smart contracts bring automation to consent management by dynamically tracking and enforcing data-sharing permissions. This shift empowers data owners to grant, audit, or strip away access permissions whenever they need to. For cryptographic security, Attribute-Based Encryption (ABE) ensures that a participant can only decrypt files if their verified attributes satisfy the access criteria. Depending on the setup, decentralized workflows leverage two unique approaches: CP-ABE embeds the access policy directly within the encrypted payload, while KP-ABE binds the policy attributes right to the user's private key. Finally, by pushing cryptographic accumulator updates on-chain, the system gains real-time revocation power. Administrators can terminate access instantly, completely bypassing the need to re-encrypt the underlying data.

On-Chain Access Control Lists (ACL) via Array Contracts

Array Contracts allow an On-Chain Access Control List (ACL) to operate as an immutable, blockchain-anchored gatekeeper. This setup dynamically manages user permissions, protecting restricted state variables and controlling access to off-chain computational pipelines. Under the hood, smart contracts leverage identity registries and key-value mappings to keep track of everyone. These maps index the wallet addresses or cryptographic public keys of approved participants, ensuring only authorized data providers, model orchestrators, or trusted execution environments can interact with the system. When a participant initiates an

operational request—such as submitting an AI model update or requesting access to encrypted datasets—the smart contract intercepts the transaction. Structural conditional modifiers then scan the array or mapping to verify the sender's signature and cryptographic clearance level before execution. If the contract authenticates the address on-chain, the transaction succeeds, allowing the system to securely orchestrate downstream processes like Proxy Re-Encryption or hardware enclave tasks. Otherwise, the contract instantly reverts the transaction to block unauthorized state manipulation or data poisoning attempts. While this approach creates a permanent compliance audit trail by logging all permission modifications indelibly to the shared ledger, developers must strategically balance its implementation with architectural optimization. Specifically, substituting raw, loop-intensive sequential arrays with constant-time mappings minimizes transaction gas overhead during high-throughput verification cycles.

Zero-Gas Verification Architecture

The Zero-Gas Verification Architecture strips away high transaction overheads by decoupling heavy computational payloads from on-chain state updates. Under this economic design pattern, the blockchain doesn't process complex operations natively; instead, it acts strictly as an immutable trust anchor to store static cryptographic evidence, commitments, or root hashes. The actual computation happens out-of-band in an off-chain pipeline. This includes validating linear (t,n) secret shares using client-side Lagrange interpolation, querying \mathcal{O}(1) constant-time mapping registries, or evaluating hardware-enclave cryptographic quotes in confidential AI setups. External nodes interact with this layer through read-only EVM view or pure functions run entirely client-side. This framework allows instant, infinite data validation without ever touching global state variables—completely bypassing network consensus gas fees.

Constant Gas Scaling under High Concurrency

To preserve decentralized network performance under intensive operational conditions, systems implement constant gas scaling to maintain stable, predictable transaction costs. This optimization pattern benefits smart contracts irrespective of total user volume or real-time transaction throughput. In traditional smart contract configurations, gas complexity scales linearly relative to the parameter n. The underlying execution layer depends on sequential loops or linear operations, frequently causing gas exhaustion failures during periods of high concurrency. Constant gas scaling eliminates these linear dependencies by substituting dynamic sequential data structures with constant-time hash tables and multi-level nested mappings to handle core protocol tasks, such as identity registration and state permission. Consequently, the computational steps required for transaction verification remain constant regardless of input size. By establishing a flat complexity layer for state logic, decentralized applications can process millions of concurrent transactions without placing undue stress on the block limit as the network scales, enabling enterprises to deploy highly scalable architectures.

14.8 BLOCKCHAIN-AI INTEGRATION FRAMEWORKS

Integrating machine learning models with distributed ledgers demands custom architectural design pipelines. A production-grade, end-to-end privacy-preserving processing pipeline typically operates according to the following sequence:

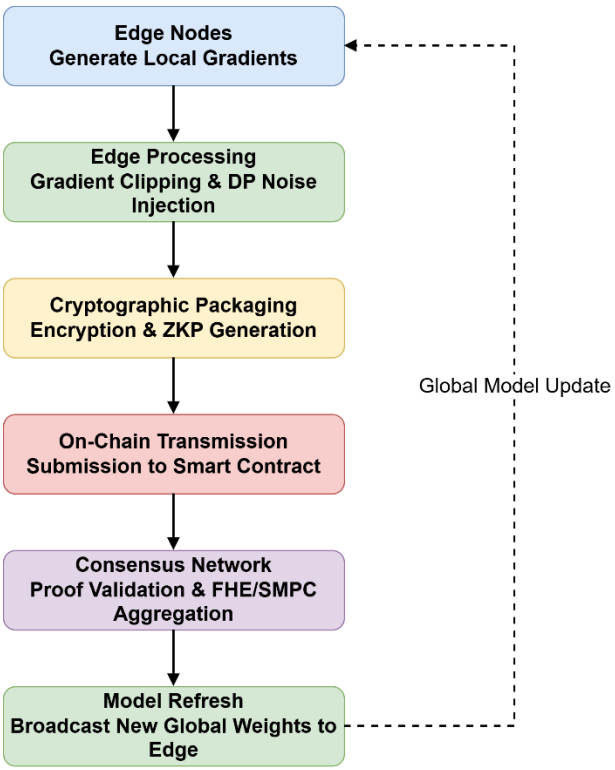


Figure – 1: End-to-end privacy-preserving processing pipeline

Modern distributed ecosystems offer highly flexible options for constructing decentralized protocols by pairing Cosmos-based blockchains with customized smart contract environments. The core Tendermint consensus engine utilizes a Byzantine Fault Tolerant (BFT) algorithm to sustain a dependable network of validating nodes, ensuring that all state transitions remain deterministic and cryptographically proven. Concurrently, the CosmWasm smart contract environment executes compiled WebAssembly (Wasm) binaries within a sandboxed virtual machine, enhancing security, interoperability, and contract composability. Leveraging Cosmos-based technology allows applications to delegate resource-intensive computations to external clusters while maintaining full control over on-chain state modifications, making it highly effective for managing permissioned, auditable validator networks.

ParaTime Decoupling Layering

ParaTime decoupling introduces a modular blockchain architecture that splits network responsibilities into two distinct layers: a foundational Consensus Layer and an isolated ParaTime Layer. The Consensus Layer handles basic network validation and state synchronization, while the ParaTime Layer isolates computationally intensive transactions and complex application logic from the main chain. This separation allows multiple independent runtimes—or ParaTimes—to operate concurrently without competing for resources. Each ParaTime can be tailored to specific operational requirements, enabling developers to integrate distinct features, such as the Ethereum Virtual Machine (EVM), customized committee sizes, or hardware-enforced TEE computations, without compromising the core blockchain's throughput or security. In a decentralized AI architecture, smart contracts efficiently coordinate operational tasks, but the underlying blockchain cannot natively process heavy machine learning workloads or maintain data confidentiality. Integration frameworks address this limitation by offloading execution tasks to off-chain nodes equipped with Trusted Execution

Environments (TEEs), such as cloud validators utilizing Intel SGX. This configuration ensures that the blockchain can cryptographically verify that the machine learning model was executed accurately inside a secure enclave before distributing protocol rewards. For resource-constrained edge architectures and IoT devices, the framework deploys ARM TrustZone to establish similar hardware-enforced protection zones. At the aggregation layer of privacy-preserving federated learning, blockchain nodes function as data providers or cryptographic aggregators. Here, Linear (t,n) Secret Sharing with Cheating Detection serves as a foundational mathematical tool, enabling nodes to aggregate model weights or gradients securely without disclosing raw underlying data. By deploying an active verification model, the aggregator can mathematically audit individual secrets during reconstruction, allowing it to rapidly identify and isolate malicious nodes that submit distorted data. This verification model minimizes computational waste by preventing the system from processing corrupted inputs. Furthermore, it insulates the smart contract from targeted data poisoning attacks where adversaries attempt to degrade the global model using fraudulent parameters. Proactively detecting anomalies during secret reconstruction preserves data integrity and optimizes network operating efficiency, allowing federated networks to maximize security while preserving structural performance.

14.8.1 Layered Reference Architecture Reference

Production-grade systems typically integrate tools from multiple domains into a cohesive, layered architecture to balance computational trade-offs. Table 1 details a representative multi-layer reference architecture.

Table 1: Layered Reference Architecture for Privacy-Preserving Decentralized Intelligence

Layer	**Function**	**Representative Privacy Mechanisms**	**Threats Addressed**
Data Layer	Local data generation, storage, and preprocessing at the edge or organizational boundary.	Local differential privacy, data minimization, metadata k-anonymization.	On-chain data exposure, regulatory compliance tension.
Computation Layer	Local model training, gradient calculation, or smart contract logic evaluation.	Secure multi-party computation (SMPC), homomorphic encryption (FHE), trusted execution environments (TEEs).	Model inversion, membership inference, computational exposure.
Coordination Layer	Aggregation of model parameters, consensus tracking, transaction ordering.	Federated learning with secure aggregation, zero-knowledge proofs (ZKPs) of correct execution.	Gradient leakage during aggregation, transparency–confidentiality conflicts.

14.9 CONCLUSION

Privacy-preserving mechanisms are critical to ensuring the structural sustainability of decentralized intelligent systems as they navigate the continuous tension between data visibility and confidentiality. Securing distributed machine learning architectures against advanced gradient leakage and model inversion exploits requires a holistic defense-in-depth framework rather than isolated patches. From a cryptographic and statistical perspective, combining Fully Homomorphic Encryption (FHE) and Secure Multi-Party Computation (SMPC) protects local parameter updates during the aggregation phase, while Differential Privacy (DP) introduces calibrated statistical noise to prevent machine learning models from memorizing confidential source attributes. To maintain systemic integrity, Zero-Knowledge Proofs (ZKPs) empower smart contracts to execute trustless verification of off-chain computations. This coordination layer is further reinforced by active auditing protocols, such as Linear (t,n) Secret Sharing with Cheating Detection, which proactively isolate malformed or adversarial parameter updates. Comprehensive environmental security is achieved by blending identity-obscuring self-sovereignty with robust physical isolation, deploying process-level Intel SGX enclaves to secure cloud validators alongside dual-world ARM TrustZone configurations for resource-constrained edge hardware. By turning immutable ledgers into secure, compliant audit logs that satisfy global data protection frameworks, this multi-layered strategy establishes a robust foundation for future research dedicated to minimizing computational latency and bandwidth overhead.

CHAPTER 15

TRUSTWORTHY AI: FAIRNESS, TRANSPARENCY, AND ACCOUNTABILITY

Contributed by: Suparna Bandyopadhyay, Swagata Das, Lipika Mukherjee Pal, Bijaya Banerjee

15.1 INTRODUCTION

The main advancement of digital technologies has brought about a paradigm shift in the way societies manage information. The transition from traditional to intelligent ecosystems has led to the tremendous evolution of developed technologies such as artificial intelligence, cloud computing, the Internet of Things, edge computing, and next-generation communication networks over the last two decades. Intelligent ecosystems are no longer involved only in the collection and storage of data, but they are able to sense the environment, learn from their experiences, adapt to external changes, and make independent decisions. This means that the intelligent ecosystems are the key factor behind the creation of a digital transformation across different industries, including health care, transportation, manufacturing, finance, education, agriculture, and governance. Such developments will allow computer systems to process significant volumes of data, predict trends, and perform calculations requiring human intelligence. They are not limited by the questions of efficiency and effectiveness, but rather include aspects of transparency, accountability, safety, privacy, human supervision, and ethics. The ability of the system to provide explanation for its decision-making, protect itself from manipulations, preserve the privacy of the individual, and maintain justice becomes essential once AI affects society to a great extent in such areas as medicine, economics, employment, legal procedures, and politics. Creation of digital trust is a mammoth task required to build trusted AI. In the past, trust came from personal relationships, institutions, contracts, and rules. Today, in decentralized digital environments, people often don't know each other or trust each other beforehand. Intelligent ecosystems usually include many organizations, software agents, devices, cloud systems, and users from different places and backgrounds.

It is important to believe that trustworthy AI must be capable of (1) making transparent decisions; (2) resisting manipulation; (3) protecting privacy and maintaining data integrity; and (4) providing equal treatment of all people who receive or are influenced by that decision. All these requirements are especially crucial in such spheres as health care, finance, the job market, law, politics, etc., where AI decides on the way people live their lives. Establishment of digital

trust is an essential task in achieving the objective of creating trustable AI. In the past, trust was created on the basis of interpersonal relations, institutions, agreements, and regulations. However, in today's decentralized digital world, people do not always know and do not trust in advance. Intelligent ecosystems generally include many organizations, software agents, devices, cloud systems, etc., as well as many users who come from different locations and have diverse backgrounds. Because of these reasons, the centralized trust model may be a bit slow today. Players need to come up with methods to check purity and integrity of certain data. But there are many benefits associated with employing a blockchain framework. Blockchain technology was originally designed for data management but now it is used to distribute trust. Recording all network transactions with distributed ledger technology. Blockchain allows trust to be created without said authority. This is done by the cryptosystem, consensus and decentralization. It is particularly valuable in intelligent ecosystems in which multiple participants collaborate but at the same time maintain the control and confidentiality of their private data. Blockchain technology coupled with AI is one such huge technological leap forward. This is a participant in a secured ecosystem, the following items have to be done: identity verification, data integrity, accountability, and setting up guidelines for governance. To be a part of a system you need to do a few things: verify who you are, keep the information safe, be responsible for your actions and set up rules for how things are run. There are ways to make this happen but using blockchain technology is one of the best ways to do it. Blockchain was first used to help people trade money like with bitcoin. Now lots of organizations use it to help people trust each other without a central person in charge. At its core blockchain is a way to keep a list of all the things that happen on a network. It is safe and cannot be changed. Blockchain uses codes, agreement rules and a system that is not controlled by one person to make sure people can trust each other even when there is no one in charge. In systems, where different people need to work together and keep their information private it is very important to have trust. Putting blockchain and artificial intelligence together might be the thing that has happened in technology so far. It is hard to turn these ideas into complex digital systems. One way to solve this problem is to use blockchain because it gives us ways to control, check and verify how artificial intelligence is used throughout its whole life cycle. Companies can take advantage of chances by using artificial intelligence that they can trust and blockchain helps make that happen. For companies that incorporate AI into their business model, they must show transparency to the customer, the regulatory body, the investor, and the public that their AI systems are as below: (1) transparent; (2) secure the data of the AI system; (3) not biased or discriminatory; (4) product liability can be controlled.

Auditing Blockchain can verify that the AI systems are in compliance with applicable laws and provide significant reductions of the risk of litigation; therefore, building trust of an AI system with customers, regulatory authorities, and investors will be much easier. Because of the decentralized nature of Blockchain, it will create an environment of innovation, as it provides a secure means of sharing data and developing team models across multiple organizations, without having to disclose proprietary information. The integration of AI and Blockchain will also create new challenges. A few examples of issues that a Blockchain may encounter in the future as they grow in popularity include: limited scalability; high computational costs; difficulties with interoperability; governance obstacles; etc. Bias, model shifts, attacks, and limited explanations are just some of the challenges faced by AI systems. We will require

cooperation from a number of disciplines in order to address these challenges. Our future intelligent eco-systems should be designed taking into account the technical and social angles so that we strike the right balance of innovation, accountability, and trust. This Top-Level Chapter has discussed how Blockchain can help establish secure and trustworthy intelligent ecosystems by promoting Fairness, Transparency, Accountable AI and Ethical AI. It will also examine the role of Blockchain as it relates to data traceability, creating trust without a central authority, auditability, automated control and secure exchange of information. The chapter will discuss opportunities, challenges and future investigations related to blockchain-enabled AIs in order to illustrate ways in which emerging technology can support the building of intelligent ecosystems that are Efficient, Transparent, Accountable and Socially Responsible.

15.2 DIGITAL TRUST IN AI-DRIVEN ENVIRONMENTS

Intelligent ecosystems represent a new paradigm of the digital world today; they are no longer isolated computing platforms, but rather combine many interconnected computing environments that can learn, adapt, and self-determine actions autonomously. Instead of previous approaches that concentrated mainly on storage and computation of transactions over data, intelligent ecosystems rely on a technological foundation including the fields of sophisticated computation intelligence, sensing, communicating and controlling the interaction of numerous connected participating entities (human/machine) of an ecosystem. Every single participating component interacts in dynamic way with other participants to fulfill common purposes. One significant characteristic of these systems consists of its capabilities in extracting meaningful data from the massive available raw data generated by every single constituent, data which support adaptive, context aware behavior. Development of such systems are supported by a lot of different interacting technological areas.AI is crucial in terms of having the intelligence to identify patterns, predict, analyze and support decision making process. Large volumes of data are manageable through cloud services which is extremely useful in supporting those systems. Also, local data processing, provided by edge computing capabilities, support decision making right at the edge. Wireless technologies, such as 5G and the new emerging 6G standard enable to maintain live connections between every geographically located component of an intelligent system. Decentralized government models enabled by blockchains provide to ecosystems' participants transparency and accountability. In these systems, data flow back and forth, fostering collective intelligence and self-adapting behavior.

15.2.1 Intelligent Ecosystem Architecture

In terms of design, Intelligent Ecosystems have a fundamental difference in architecture when compared with traditional centralized information systems. Conventional architectures typically utilize a hierarchy (pyramid) in which data is accumulated through a user; transmitted to a data center (server); processed by the organization that owns the server (and may or may not have value); and delivered back to the user. These basic building blocks including various architectures can only perform simple and routine operation activities. Unfortunately, their capabilities are ill-suited to the new demands that have emerged in today's complex, scalable, and always-on digital environment. The Intelligent Ecosystem will be implemented on the basis of the latest distributed and/or multi-layered architectures that allow for decentralized decision-making, continuous communication and rapid responses to emergent situations.

Another dimension of the Intelligent Ecosystem is the sensing/data gathering layer, which concerns the data collection process from physical and digital sources. This layer adds a new functionality to the system. The sources can be various, from industry and environmental sensors, corporate information systems and social media to IoT devices and wearables.

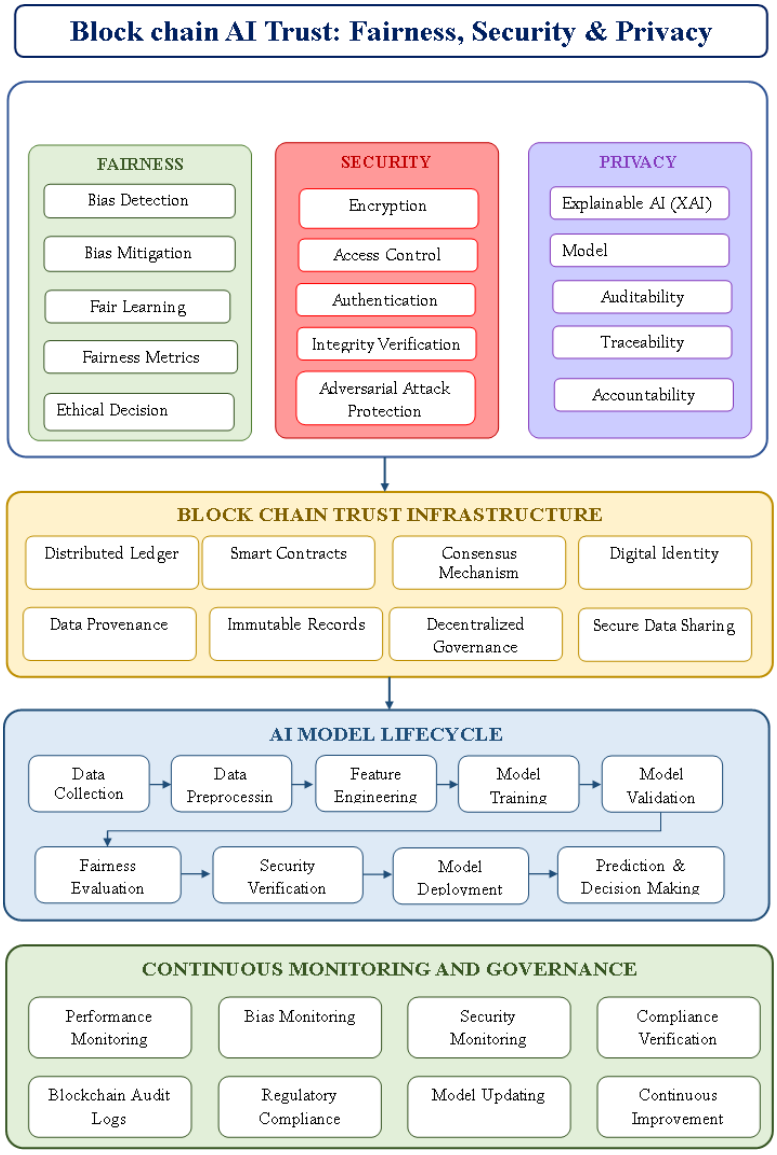


Fig 1: Block diagram of Blockchain AI fairness, trustworthiness, privacy

The data obtained from these sources also varies in its reliability and frequency. The communication layer, which lies above the sensing layer, allows for efficient and safe communication among the participants of the ecosystem. The communication layer uses networking technologies and protocols and standards that enable interoperability to establish seamless connections between a variety of devices. The second part of the ecosystem is the processing layer. This layer is responsible for converting the initial information into knowledge. Generally, cloud computing services are providing a large amount of computing power. Edge computing allows localized computing for less latency. On top of these layers, the intelligence layer is built, which contains artificial intelligence systems that can think, predict, optimize and make decisions by themselves. Finally, the governance and trust layer is in charge of managing identities and establishing rules and laws to keep an eye on the ecosystem. The majority of the

time, blockchain technologies are crucial in creating these layers because of the data's dependability, the decentralization of governance, and the opportunities for auditing. These changes have an effect on the environment, creating new data for next learning cycles. Human actions change the environment and generate new evidence that will be used in the further cycles of learning. Such a loop allows ecosystems to evolve due to repetitions and inductions, or inductions and repetitions. However, such an approach poses challenges in terms of model drifts, feedback, and undesired outcomes.

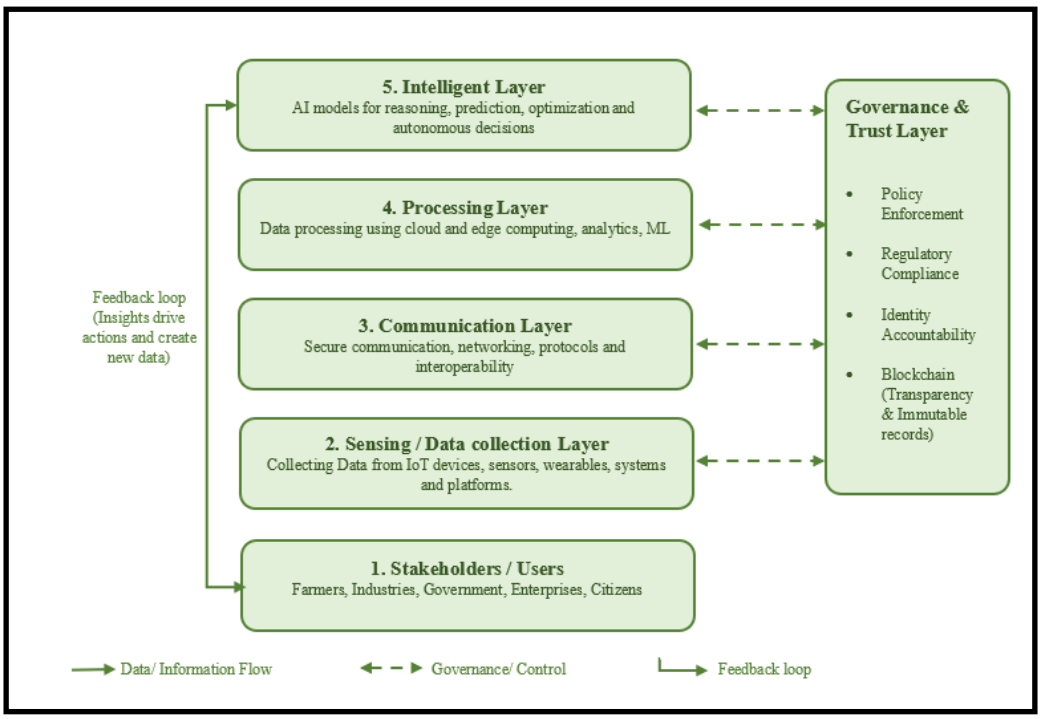


Fig 2: Basic Architecture of an Intelligent Eco-System

15.2.2 AI-Enabled Digital Transformation

The rise of artificial intelligence has been a major catalyst driving the digital revolution of both current day societies and organizations. The original emphasis of digital transformations was on changing manual processes into computerized processes; improving the efficiency of operations and access to information were the primary goals. With the introduction of technologies that incorporate artificial intelligence, this vision has dramatically expanded by providing systems with not only the ability to automate processes, but also to learn from their experience, interpret complicated situations, and assist with developing strategic decisions. Therefore, AI-enabled transformation changes the focus of digital transformation from one that considers only process automation to one that incorporates both process automation and cognitive transformation, whereby machines contribute to intelligence and innovation within an organizational setting. The effects of AI-enabled digital transformation can be observed throughout an array of industries. In the area of health care, machine learning has been able to provide clinicians with assistance in diagnosing diseases, predicting patients' outcomes and personalizing treatment methods. The world of finance utilises analytics developed utilizing AI to determine risk, identify potential instances of fraud, and optimize investment opportunities. Moreover, the manufacturing industry has embraced predictive maintenance technology that helps them predict any machine malfunctions hence improving their operational dependability.

Learning platforms utilize adaptive learning technologies that tailor educational experiences to suit each learner. Many examples exist where AI may not just increase efficiency of existing processes but would allow organizations to transform their core processes of value creation and service delivery. However, ethical and governance issues exist as a result of the impact of AI-driven digital transformation. AI systems are usually very sophisticated computational algorithms that cannot be easily understood because of the difficulty in comprehending the reasoning process used in the development of such a system. In addition, deep learning algorithmic models offer exceptional predictive capabilities but fail to give transparency on the process of reasoning involved in making predictions. The absence of such transparency leads to lack of trust among other issues that would be experienced in domain areas where the use of the AI system has significant effects like in healthcare, employment, public policy, and legal decision-making. Therefore, future processes of digital transformation will have to incorporate explainability and accountability mechanisms. Data dependency can be a huge problem - AI systems require training datasets that reflect past circumstances and practices of the organization. As a result, the models it produces are likely to reinforce inequality if the data sets are biased, inaccurate, or unbalanced. Therefore, when you transform into digital, make sure you manage data. Using blockchain technology, which provides an accurate, unchangeable, and verifiable history of the data used in the digital transformation process, is one way. Decentralization! In a single organization, there is one organization that controls resources for calculation and decision-making. Although for a single organization it might be easy to maintain everything, there are dangers of concentration, scaling, and single points of failure. Decentralized computing theories are based on the discipline of distributed systems, which studies how geographically dispersed entities can cooperate to reach a common objective. In decentralized computing, all nodes will contribute to processing power, storage, or analysis capabilities without being connected. For coordination among the nodes, I would use different techniques such as communication strategies, synchronization techniques, and consensus algorithms that would allow the nodes to reach their decisions without any central authority telling them what to do. This makes the nodes stronger in a decentralized network since they don't depend on the infrastructure they operate on. Recent advancements in blockchain technology have created increased interest in decentralized computing as they provide practical examples of how to establish trust among users in a way that does not require either of them to have complete trust in each other. Through the use of distributed ledgers and consensus mechanisms, blockchain networks are capable of maintaining a consistent and properly sequenced copy of the information that each node contains. Additionally, the use of distributed ledgers and consensus mechanisms will help ensure the integrity and validity of the data recorded on a blockchain have been recorded accurately. The use of consensus mechanisms can put a strain on computation, it can become harder to coordinate governance processes, and it can be difficult to establish worldwide interoperability of various systems. Therefore, intelligent ecosystems of the future will probably be based on hybrid systems that provide centralized efficiency and decentralized trust and resiliency.

15.2.4 Distributed Intelligence

Distributed intelligence expands upon the idea behind decentralised computing by using a wide range of cognitive capabilities not contained within any AI platform but rather by distributing

them across an ecosystem. Several autonomous agents participate in the learning, reasoning, and decision-making. Intelligent sensors, edge devices, autonomous cars, robotic systems, software agents, and cloud-based analysis tools are a few examples of autonomous agents. Intelligence is distributed, and decisions are made nearby. For instance, autonomous cars must make decisions based on data from the surrounding environment. It will take time to analyze the data and make the necessary decisions to ensure safety if these vehicles rely on a central system that is far from here. You will still be able to coordinate with the rest of the world around you, by using scattered intelligence. Federated learning (FL) is an example of distributed intelligence. In the context of FL, each device involved in the training procedures develops the model locally and only transmits the updates for the central model back to the server, eliminating the need to send sensitive data to a centralized server for model training. This makes it possible to preserve privacy and facilitates teamwork during dispersed training. By offering safe coordination, transparent contribution tracking, and decentralized governance, blockchain technology enhances the idea of federated learning. There are certain issues with the intelligence distribution process as well. The existence of conflicting and incomplete information will make it extremely difficult for different agents to coordinate their knowledge. It's possible that each individual agent has different local optimization goals that don't match the goals of the ecosystem. To overcome these challenges, there needs to be a strong trust framework in place to facilitate secure collaboration and transparent governance.

15.2.5 Ecosystem Governance

The way in which intelligent ecosystems achieve alignment with their organizational objectives, societal values, and regulatory requirements is through governance, which is the foundational mechanism for how these ecosystems will work. The current ecosystem governance models will have to be improved if we want to see the better functioning of the system with humans, AI systems, and decentralization infrastructure in the light of developing intelligent ecosystems and their greater autonomy. These are just a few examples of ecosystem governance: Strategy, risk, ethics, responsibility, transparency, and participation. Additional governance will manage the policy in fair algorithmic practices, data ownership, data protection, security management, and accountability. Governance structures will make it easier to control who is following the current policy and settle disputes when someone doesn't follow the policy or has a bad result. With the help of blockchain technology, there's so much potential for improving our governance. Using distributed ledgers will allow us to keep a record of everything we did and what happened when we had to comply, but using smart contracts will make us sure that the rules are being followed. These features make a company more accountable and give the chance for an auditor to audit. Moreover, I expect that by having a decentralized government, all of us will be able to make local decisions, which will make everything open and transparent. Whether or not the new governance systems balance efficiency, trust, privacy, accountability, and ethics will determine how successful intelligent ecosystems continue to be. The achievements and innovations, like the development of artificial intelligence or blockchain technology, are going to influence leader performance in the intelligent ecosystem and become the main factor that will determine whether intelligent ecosystems will gain legitimacy.

15.3 DIGITAL TRUST INFRASTRUCTURE

Since everyone will be interacting in a variety of settings where technology is crucial, fostering trust among people will provide positive outcomes. Working with traditional organizations, which establish trust through contractual relationships, hierarchical structures, and mutual knowledge, is very different from this. Participants in intelligent ecosystems will not know one another, but they will engage with technologies such as machine learning and autonomous software agents in addition to people. In such a scenario, the development of safe, dependable, transparent, and responsible exchanges and transactions in the increasingly decentralized world would depend heavily on digital trust. The level of trust that stakeholders have in digital technology, systems, and governance used to conduct business in a way that is safe, morally correct, predictable, and compliant with specific standards is known as "digital trust." Data integrity, privacy, openness, responsibility, fairness, dependability, and governance are just a few of the many facets of trust in intelligent ecosystems that go beyond cybersecurity concerns. Establishing the foundations of digital trust is crucial given the growing automation and impact of intelligent technology on critical business processes and decision-making. Businesses and societies alike are understanding that innovation alone will not be enough for technologies to be sustainably accepted; instead, there must be a focus on developing ways to demonstrate reliability through verifiable evidence and openness in behavior. The majority of automated transactions do not include a someone who oversees the transaction, which is a problem with digital trust in general. Many intelligent systems process information and make judgments without constant human interaction. Examples include autonomous cars traversing cities, machine learning algorithms determining a person's credit eligibility, and predictive analytics systems offering patients treatment recommendations. As a result, all parties involved in such instances need to have a reasonable level of confidence in the underlying technology, as well as the methods used to collect, analyze, and use the data. This need for increased confidence has led to significant amounts of research into developing digital trust infrastructures that can facilitate trustworthy transactions for all stakeholders involved throughout the entire lifecycle of an intelligent system.

15.3.1 Digital Identity as the Foundation of Trust

The foundation of the digital trust asset infrastructure is trust. In order to build trust in the physical world, we can rely on many methods such as relying on observations, legal documents, affiliation with an institution (like a college or a government), or developing social relationships with other people, etc. In the digital world, we must find similar methods to help verify identities while maintaining the rights of individuals to maintain their privacy and secure their data. For all forms of communication (e.g. people to people, people to devices, people to applications, people to services, people to autonomous agents), every member of the group should have a fully verified digital identity prior to engaging in any form of communication (within the group and between two or more groups). The traditional approach to managing digital identities uses centralized authorities to issue, store and validate credentials issued within the digital identity management system. The centralized authorities are Government issued digital identities, Organizational Authentication Systems, Commercial Identity Services) The traditional model currently being used to manage digital identities provides many benefits, however, there are numerous areas in which the use of centralized identity repositories leave

much to be desired. Regarding the storing, exchange, and usage of user identifying information, there is absolutely no control. Individuals have relatively little control over how their identity is used, shared, and stored. New opportunities to improve users' control over their identification information are now being presented via decentralized identity systems. Users with self-sovereign identities have control over the history of their credentials and can reveal their data if needed. These solutions leverage blockchain technology, which enables identity verification and credential recording without centralization. Second, individuals can use cryptographic technologies to demonstrate that they are the owner of certain traits (such age, citizenship, and professional credentials) without disclosing any personal information. Users find it simpler to defend themselves, their privacy, and against dishonest behavior. Even though autonomous interactions occur in the context of AI agents, identity management plays a crucial role in this situation as well. The basis for distinguishing between agents who are legitimate and those who are not is verifiable identities. Digital identity serves as the basis of all trustworthy systems.

15.3.2 Trust Models in Intelligent Ecosystems

Trust in intelligent ecosystems is not a binary notion but a multi-dimensional construct, shaped by technical, social, organizational and behavioral factors. In order to gain an insight into the ways in which trust may be generated, assessed and managed in the digital age, researchers have proposed various types of trust models. There are three major types of trust models, namely institutional trust, reputation-based trust, and computational trust. The concept of institutional trust rests on the presence of some organizations and their corresponding laws. Individuals tend to trust government, healthcare and financial institutions since they are controlled and regulated. However, institutional trust cannot work in a distributed environment where relationships take place between actors that operate beyond organization boundaries. The concept according to which previous behaviors can serve as an indicator of trustworthiness is termed as reputation-based trust. In social networking sites, virtual markets, and cooperative networks, reputation is used as a means to facilitate trust among participants. While this mechanism works great, it is extremely vulnerable to information asymmetries, collusion, and manipulation. Another far more sophisticated way is the computational trust, which uses algorithms to create trust relationships based on observations. Transaction history, behavioral patterns, credentials, compliance, and performance are among the criteria that influence the trust estimation process. Artificial intelligence is increasingly gaining importance in computational trust systems through analyzing data and identifying anomalies to determine trustworthy behavior. Since blockchain technology allows for generating immutable proof of the trust-related transactions, it is another step towards building an efficient computational trust system. Integration of various trust frameworks becomes essential due to the complexity of the intelligent ecosystem. Sufficient trust system should include technical verification, governance, and social accountability mechanisms. Both technical and organizational aspects of the trust concept are recognized by multidimensional approaches.

15.3.3 Cryptographic Trust and Data Integrity

Cryptography stands out as one of the most important methods for digital trust. The techniques that guarantee the confidentiality, integrity, authenticity, and non-repudiation of digital information transmission and transactions are offered through cryptography. All of them act as

the backbone that guarantees the legitimacy of the communication process in the ecosystem. As intelligent systems depend on good data, the issue of data integrity becomes even more significant when talking about intelligent ecosystems. Both the training data and data that is used as an input into the system act as the basis for prediction or behavior generated by the machine learning algorithm. In such cases, incorrect predictions or malicious activities will happen if the bad data is used to predict the behavior of such models. Hashing helps us to authenticate our data. Cryptographic hashing functions take any kind of input data and produce a string of bytes of fixed size. This is going to be unique for that particular input data. Users can compute the hash value of their data and distribute the hash value of the data. The receiver can then compute the hash value of the data he receives and make sure it corresponds to the initial hash value. In this way, they can verify that the data has changed in case of mismatch. Digital signatures will guarantee that the information you receive originates from a reliable source. When asymmetric encryption is used, the sender's private key is used to sign the data, while the recipient's public key is used to verify it. Data source authentication and data integrity will be guaranteed by this method. Systems that can trust one another are necessary in AI applications where numerous firms supply data, algorithms, and processing capacity. Cryptographic trust can be implemented in a distributed database thanks to a technique called blockchain. Transactions that are cryptographically connected to earlier blocks of transactions are contained in blocks on the blockchain. The data cannot be altered once it has been verified and added to the blockchain.

15.3.4 Privacy-Preserving Technologies

ML model decisions rely on data passed into the model (and data model was trained on), so passing in a bad set of inputs can lead to some false predictions or even the generation of fraudulent activities. By using cryptographic functions, we can be sure that our data cannot be tampered with. Hashing helps confirm the integrity of data. A hash algorithm takes an input and produces a unique string of bytes of fixed length for each unique input. Even a minor change to an input will result in a very different output hash. You can hash data and share that output so the receiver can hash the content they receive to verify its integrity. Digital signatures can prove where certain information comes from. With asymmetric key pairs, you can use a private key to digitally sign data then use corresponding public key to digitally verify the signed data. Take piece of data hash it, sign that hash of data with private key and send it over to the other party where they take their data, sign that hash with their own private key then they take that signed data then and take hash of that signed data then compare it to the originally provided hash value and they know if any of the data was altered and also this ensures the source of the data too. Distributed Machine Learning Systems and Blockchain When it comes to AI, a lot of times there are many different parties who might have their own input on several AI algorithms and machines, as all those machines and inputs are often relying on one another, you are gonna want a framework of trust there for your distributed data/ML/computing environments. Blockchain has a great capability of bringing this idea of cryptographic trust into the distributed database tech. The technology consists of links of transaction blocks, where each one references the hash of the one previous. The distributed data/ML/computing platform that blockchain gives ensures immutability and transparency after its validation. As such, along with its capacity for

the immutability and transparency of data, blockchain directly enables provenance, quality assurance, accountability, and auditability.

15.3.5 Blockchain-Based Trust Management and Governance Implications

Blockchain has emerged as one of the most impactful innovations to date in digital trust management. Blockchain is different from the way of doing things. Usually, we need someone in the middle to make sure everything is okay. Blockchain does not work like that. It uses a kind of agreement that everyone agrees on. This way we can trust that everything is correct. In places blockchain is like a layer that keeps us safe. It keeps track of what happens with our money. Who we are. It also makes sure that people are who they say they are. It keeps an eye on who owns what. When something happens with our money it gets written down in a book. This book is called a ledger. What gets written down in the ledger stays there forever. We can always go back. Look at it. his is good for intelligence. We can see what happened with the information and how decisions were made. Blockchain is like a paper trail that we can follow. It helps us make sure that everything is fair and okay. Blockchain is important for blockchain because it helps us trust the system. Blockchain and blockchain systems need to be transparent. Blockchain is the key, to making this happen. Smart contracts make blockchains better. They give a way to manage processes and transactions that can be programmed. They are codes. These codes start actions when certain conditions are met. With contracts people can set rules. They can see who can see data. They can check if others are following the rules. They can keep track of decisions. This reduces mistakes and cheating. Everything happens automatically. Smart contracts help to prevent errors and wrongdoing. They remove the need for people to enforce the rules. Smart contracts are helpful. The problems that blockchain technology is facing today are as follows:--

- scalability
- interoperability
- energy use
- governance

Morever, technology just can't replace ethics, reasoning and responsibility. We need a mix of people and technology along, with laws, policies and institutions that enforce these policies. This combined approach will help establish trust. As the intelligence ecosystem gets more complex understanding trust architecture will be really important. In the future researchers will likely focus on creating frameworks that're universal and work well with different systems. These frameworks should also be able to grow with the ecosystem respect peoples' privacy and follow regulations. The goal is to make sure digital trust architecture is a part of the intelligence ecosystem.

15.4.1 Data Monopolies and Concentration of AI Power

Data monopolies are a problem with centralized AI solutions. The main issue is that current AI models need a lot of data to work and get better. Companies that have a lot of data have an advantage. This is because having different types of data helps an AI system succeed. Big tech companies have collected a lot of user data from things like search engines and social networks and online stores and smartphones and cloud services. This has made the AI world unfair, where a few companies control the computers and data needed to make the next generation of

artificial intelligence, which is the next generation of AI and these companies have a lot of power, over AI. Public organizations, smaller companies, and academic institutions are often sidelined because they do not have similar access to computational resources. Concentrating data ownership can lead to the following problems: -

1.**Restricting Innovation**: Data concentration places hurdles on newcomers into the field of AI, as there is insufficient computational power for training.

2. **Creating Dependencies**: The market relies on these service providers for the development and provision of essential AI applications. 3. Eliminating User Control: Individual users often generate enormous quantities of valuable data without having ownership or control over how that data is used.

Such unequal control of information resources by only a few actors can also give rise to inequalities in our society—communities generating huge amounts of digital information, for instance, derive only a small share of its value. It has also fueled the discussion around how decentralized ownership can empower both individuals and companies to retain control over their information and at the same time help facilitate collaborative AI development. Decentralization in AI is, as a result, an endeavor that aims at mitigating the above challenges by distributing control over data, processing power, and intelligence training among several actors. Techniques such as federated learning, distributed ledger systems, and peer-to-peer networks enable developing AI capabilities without any centralized storing of information.

15.4.2 Algorithmic Opacity and Lack of Transparency

The other problem, like algorithmic opacity—also called the "black box problem"—is another key issue with centralized AI, following as:

- The decision-making logic behind most advanced AI models, including deep-learning ones, often consists of billions of interrelated parameters and decision points.
- While these models can achieve highly impressive accuracy on various predictive tasks, even their own creators may struggle to understand precisely why they arrived at their output or decision.

In the context of centralized AI, a private entity usually governs how and what algorithm drives the results and has incentive to keep the details secret for intellectual property and commercial interests. In this paradigm, the individual may get a product recommendation or a medical prognosis, but has no visibility into how the factors they contributed were weighted or why this result was recommended over any others. Algorithmic opacity becomes a major barrier to adoption and acceptance when AI decisions can determine whether or not one qualifies for a loan, gets hired for a job, receives medical care or is convicted for a crime. The difficulty of validating the trustworthiness of an AI model can erode public confidence: If an automated system says no to one's request for credit, one doesn't always know if that was due to a lack of financial viability or an embedded bias in the algorithm. This obscurity can be equally confusing for scientists and policymakers. Without adequate access to model architectures, data characteristics, and decision-making procedures, independent audits are challenging and, as a result, errors can go undetected and processes can lack fairness and ethical adherence. However, decentralized AI paradigms could provide opportunities by enhancing transparency

through, for example, decentralized model verification, common protocols, interpretable AI mechanisms and collective decision-making.

15.4.3 Unfairness and Biased Decision-Making

The process through which AI systems make predictions is to learn from data collected in the past. This makes possible a variety of advanced capabilities, but it is precisely because AI models learn from past data that they may duplicate or intensify already existing socio-economic and cultural inequalities. Bias in centralized AI systems, that tend to be managed by a specific small group of people for a limited goal, can exist in various steps, such as data collection, data labeling, feature selecting, algorithm designing, or deployment strategies. When the input of AI systems include disparity represented dataset and underrepresented peoples' data; a poorly represented dataset will cause the prediction to unfairly discriminate against certain groups. An AI recruitment software, which had been based on traditional discriminative hiring, will likely be prone to favoritism towards certain social groups. Or that facial recognition systems had an inferior detection accuracy for individuals in ethnic and social minorities because the database contained insufficiently representative strains. The risks in centralized AI environments increase due to concentration of decision-making power. If there's one central provider responsible for building, managing and owning the whole system, then it's quite difficult to find bias in the system, especially if it has millions of users. Decentralization in AI is critical in addressing the issue of AI bias due to the power to distribute decision-making responsibilities.

15. 5 SECURITY VULNERABILITIES AND SINGLE POINT OF FAILURES

The centralization of AI makes data more susceptible to cybersecurity threats due to large amounts of data, computation, and storage concentrated in the hands of just a few players. If a cyber attacker takes out a centralized AI supplier or provider, many private datasets can be lost, or critical AI services can cease to be available to others. Centralized repositories often present a huge target for hackers, as a breach on one site can give them access to data that will impact hundreds of thousands or millions of people. This information could be anything, from banking information, healthcare data, and personal details to organizational data, and a large-scale breach could have massive privacy or economic fallout. Another issue is the potential of the model to be subverted or compromised by outside agents via various methods, including Data poisoning attacks. The feeding of data to the model, which has been purposely doctored or malformed with the intention of sabotaging the accuracy or impartiality of the output or decision-making. Adversarial training model Manipulation of output data. The addition of subtly altered inputs, which may only alter the AI decision by 1 or two percent, is enough to have a very serious impact given large-scale operations. Model reverse engineering. Backdooring models or directly interfering with the model’s parameters or code. As centralized models typically use one system and a single pipeline to feed in training data, if that is compromised, then any and all users of that central AI system will be impacted. Operational concerns also emerge: if your AI supplier experiences technical glitches, network outages, or policy changes to its intelligent solutions, that can significantly impair your operations or lead you to the inability to use core intelligently enabled functions, as the impact of the failure of the

vendor impacts multiple enterprises, organizations, or even an entire ecosystem. This single point of failure has the potential for severe fallout on an industrial scale. Decentralized AI may use a different set of mechanisms to provide for these security requirements by decentralizing both data, computation, and decision-making among a number of different participants and utilizing concepts like blockchain-based verifications and distributed consensus mechanisms. Distributed secure learning systems are examples of security mechanisms for decentralized models.

15.6 GOVERNANCE LIMITATION(S) AND REGULATORY RISKS

Managing the development of intelligence in any specific area will be a tough job because technology advances faster than regulations can keep up.

We normally think of rules in terms of groups and organizations. Artificial Intelligence systems operate globally, meaning that such systems must take into account many users, information sources, and computer systems. Large organizations that provide intelligence services will surely develop their guidelines, concerning how to process the data using their services and algorithms. The artificial intelligence systems are applied by those large organizations. In the case when large organizations develop their guidelines, it will not be sufficient since there can be situations where the interest of the organization does not meet the interests of other people. Such artificial intelligence systems must be taken into consideration in this case. There are two main problems arising from controlling the intelligence system.

- The first problem is that a lot of things are kept secret like where the information comes from and how the system is updated and tested inside the organization.
- The second problem is that it is hard to make sure the intelligence system is doing the thing.

People who make intelligence systems will keep making rules about how to use artificial intelligence. It will be tough to manage how intelligence is used in an area because technology is moving faster than rules can be made. But AI systems are used over the world so they have to think about a lot of users, information sources and computer systems. Big companies that offer intelligence services will probably make their rules about how to handle data through their services and algorithms. When we expect these companies to make their rules it will not be enough because sometimes what the company wants is not what other people want.

15.7 CONCLUSION

Historically big centralized AI systems have driven progress. They helped deploy models and intelligent applications on a large scale. However, these systems have some problems. They are controlled by a people who own the data. This leads to issues like monopoly and much control over AI. The systems are often unclear which causes bias. Also, when data and algorithms are concentrated, they can be vulnerable to security threats. There are also rules and limited accountability. The centralized ownership and control of data cause these problems. Centralized AI systems also have a lack of transparency. This makes it hard to understand how

they work. Large centralized AI systems have driven progress. Their problems need to be addressed. The issues, with AI systems are significant. As we integrate our societies and economy ever deeper with intelligent systems, reliance on a centralized system is set to undermine equitable distribution of economic power and public faith in the system. Decentralized AI seeks to address such issues through moving towards systems where distributed intelligences, clear and open governance, privacy and data protection, and shared ownership of digital resources can foster a new path forward for trustworthy AI—a model combining efficient centralized computation with open and resistant distributed systems.

PART III

INTEGRATION, APPLICATIONS, AND FUTURE FRONTIERS

CHAPTER 16

TRUSTWORTHY AI: FAIRNESS, TRANSPARENCY, AND ACCOUNTABILITY

Contributed by: Dr. Ranjan Kumar Mondal, Jayanta Chowdury, Palash Baran Panja

16.1. INTRODUCTION

The accelerating convergence of Blockchain and Artificial Intelligence (AI) represents one of the most consequential technological developments of the twenty-first century. Traditional centralized AI systems suffer from persistent deficiencies. These include opaque decision-making processes, concentrated data ownership, and deep-rooted trust deficits. Together, these limitations impede responsible deployment at scale. Blockchain technology offers a complementary solution to these challenges. Its cardinal properties include immutability, decentralized consensus, programmable smart contracts, and transparent audit trails. Despite a growing body of research on individual aspects of this convergence, the literature still lacks a unified and comprehensive framework for Blockchain–AI integration.

Major findings include:
(1) Blockchain-enabled federated learning demonstrably improves model fairness, contribution accountability, and privacy preservation.
(2) Token-based incentive mechanisms reduce coordination failures in decentralized data markets. However, they remain vulnerable to Sybil attacks and incentive misalignment.
(3) Existing platforms such as SingularityNET, Ocean Protocol, Fetch.ai, and Bittensor exhibit significant trade-offs among scalability, governance decentralization, and privacy guarantees.
(4) Regulatory fragmentation, particularly with respect to GDPR and cross-border data sovereignty, constitutes a primary barrier to enterprise adoption.

Building on these findings, we propose the Trusted Decentralized Intelligence Marketplace Framework (TDIMF). This original five-layer architecture integrates privacy-preserving computation, on-chain governance, and tokenized incentives into a cohesive ecosystem. The paper concludes by delineating eight high-priority future research directions. These include quantum-resistant blockchain-AI architectures, autonomous AI agents with blockchain identity, and explainable AI auditing systems. The contributions of this work are intended to guide both academic inquiry and practical system design in the emerging field of decentralized intelligence

economies. Artificial Intelligence has undergone several paradigms shifts since its formal inception at the Dartmouth Conference in 1956. It has evolved from early rule-based expert systems to the statistical learning revolution of the 1990s. The field entered a new phase with the deep learning renaissance inaugurated by landmark ImageNet work. As a result, AI systems have grown exponentially in capability. The contemporary landscape is defined by large language models, multimodal generative systems, and reinforcement learning agents. These systems are capable of surpassing human performance across a widening array of cognitive tasks However, this rapid increase in capability has not been matched by comparable advances in transparency, accountability, or equitable access. Blockchain technology was introduced by through the Bitcoin whitepaper. It proposed a radically different mechanism for coordinating trust among mutually distrusting parties. The subsequent introduction of programmable smart contracts by Ethereum expanded blockchain's potential far beyond digital currency. It enabled self-executing agreements, decentralized governance, and token-based economic coordination. The maturation of Web3—the emerging paradigm of decentralized internet infrastructure—has further accelerated interest in blockchain as a foundation for trustworthy digital economies.

The convergence of these two technological trajectories has given rise to an emerging interdisciplinary field. This field focuses on deploying AI capabilities within blockchain-mediated trust environments. The convergence is not merely additive. Blockchain's structural properties, including immutability, decentralized consensus, and transparent auditability, directly address the transparency and accountability deficits of conventional AI systems. At the same time, AI capabilities can improve blockchain efficiency. They support intelligent contract optimization, anomaly detection, and predictive validation.

16.2. MOTIVATION: THE CRISIS OF TRUST IN CENTRALIZED AI

The dominance of centralized AI architectures has produced a constellation of interrelated problems. These problems constitute what this paper terms the "centralized AI trust crisis." Data ownership is concentrated among a small number of platform operators. These operators maintain unilateral control over training corpora, model parameters, and inference infrastructure. This concentration creates substantial market power asymmetries. Data contributors receive only limited compensation for the value they generate. Downstream consumers of AI services have minimal visibility into the systems governing their interactions. Model opacity presents a fundamental challenge for contemporary AI systems. Black-box deep learning models make consequential decisions in areas such as credit scoring, medical diagnosis, and criminal sentencing. However, they do not provide auditable reasoning processes. The European Union's General Data Protection Regulation (GDPR) establishes a "right to explanation" for automated decisions. Current AI systems, however, often cannot satisfy this requirement in a rigorous manner.

Privacy risks are equally pervasive. Centralized training on sensitive health, financial, or behavioral data creates persistent exposure risks. These risks remain even when individual datasets have been anonymized. Trust deficits also affect AI supply chains. When AI models are distributed through centralized repositories, there is no reliable mechanism to verify their provenance, training conditions, or behavioral specifications. Adversarial modifications may be

introduced without detection. Licensing terms may be be violated without accountability. Intellectual property contributions from model developers and trainers can also be appropriated without compensation. Collectively, these structural limitations impede the responsible scaling of AI deployment across high-stakes application domains.

16.2. THEORETICAL FOUNDATIONS

16.2.1 The Artificial Intelligence Ecosystem

Machine Learning forms the methodological foundation of contemporary AI. It includes supervised, unsupervised, and semi-supervised learning paradigms. That enables systems to generalize from data instead of relying on explicit programming. Deep Learning is a specialized branch of Machine Learning that uses multi-layer artificial neural networks. It has shown exceptional capability for representation learning. The development of transformer architectures marked a major advance in generative AI. These architectures have enabled systems to generate coherent text, synthesize realistic images, and perform complex reasoning tasks. Reinforcement Learning conceptualizes intelligence as the optimization of cumulative reward through continuous interaction with the environment. This learning paradigm supports a wide range of applications Federated Learning (FL), introduced by Google, is particularly important for the present analysis. It enables distributed model training across decentralized data silos without centralizing raw data. Under the FL paradigm, local model updates rather than raw data are aggregated by a central coordinator. This approach substantially reduces data exposure while enabling collaborative learning. The architecture is directly relevant to blockchain integration because the coordination function of the central server is a natural candidate for decentralization through smart contracts.

16.2.2 Blockchain Fundamentals

Distributed Ledger Technology (DLT) maintains synchronized records of transactions across a network of nodes. It does so without requiring a trusted central authority. Blockchain is the most widely deployed form of DLT. It organizes records into cryptographically linked blocks. This structure ensures that modifying any historical record requires reconstructing all subsequent blocks, which is a computationally prohibitive task in well-secured networks. This immutability property forms the foundation of blockchain's value as an auditing and provenance mechanism. Consensus mechanisms determine how distributed networks agree on the canonical state of the ledger. Proof of Work (PoW), as implemented in Bitcoin, requires nodes to expend computational resources to validate transactions. This mechanism provides strong security guarantees but consumes substantial energy. Proof of Stake (PoS), adopted by Ethereum 2.0, selects validators in proportion to their staked cryptocurrency holdings. It achieves comparable security while dramatically reducing energy consumption. Alternative mechanisms, including Delegated Proof of Stake (DPoS), Proof of Authority (PoA), and Byzantine Fault Tolerant (BFT) variants, offer different trade-offs among decentralization, throughput, and finality latency. Smart contracts are self-executing programs deployed on blockchain networks. They encode contractual logic in tamper-resistant code that executes deterministically when specified conditions are met. Ethereum's Solidity language and EVM

(Ethereum Virtual Machine) provide the dominant smart contract execution environment. However, alternatives such as Rust-based Solana programs and CosmWasm offer enhanced performance and security properties. Smart contracts enable automated royalty distribution, access control enforcement, identity verification, and governance voting without intermediary involvement. Tokenization is the representation of assets, rights, or utilities as on-chain digital tokens. It enables programmable economic incentives within blockchain ecosystems. Utility tokens confer access rights to platform services. Governance tokens convey voting power in decentralized autonomous organizations (DAOs). Security tokens represent claims on real-world assets, while non-fungible tokens (NFTs) establish verifiable ownership of unique digital or physical items. The design of token economic systems, known as tokenomics, is a nascent but rapidly maturing discipline. It has direct implications for the sustainability of decentralized AI platforms.

16.2.3 The Convergence of AI and Blockchain: Complementary Capabilities

The convergence of AI and blockchain is analytically grounded in the structural complementarity of their respective capabilities. AI systems excel at extracting intelligence from data. They make predictions, recognize patterns, optimize decisions, and generate novel content. However, they typically operate as opaque, centralized systems that lack inherent accountability mechanisms. Blockchain systems provide trust infrastructure through immutable records, transparent execution, and decentralized governance. However, they lack native intelligence. They can verify and coordinate activities, but they cannot learn or make predictions. This complementarity is captured in Table 1, which maps AI capabilities to the blockchain properties that can extend, verify, or distribute them.

Table 1. AI–Blockchain Capability Complementarity Matrix

AI Capability	Limitation	Blockchain Capability	Synergistic Outcome
Predictive Intelligence	Opaque reasoning	Immutable audit trails	Verifiable and explainable predictions
Automated Decision-Making	Centralized control	Smart contract execution	Decentralized, tamper-resistant automation
Learning from Data	Privacy risks in centralization	Federated coordination via consensus	Privacy-preserving distributed learning
Model Optimization	IP theft and misappropriation	Tokenized model licensing	Attributed, compensated AI contributions
Pattern Recognition	Lack of provenance tracking	On-chain data provenance	Traceable, auditable data pipelines
Generative Synthesis	Deepfake and misinformation risks	Content authentication tokens	Verifiable content provenance

The synergistic potential articulated in Table 1 is not merely theoretical. Emerging platforms are actively engineering systems that harness both AI intelligence and blockchain trust: federated learning protocols replace central coordinators with smart contracts; model

provenance is recorded on-chain; data contributors receive token-denominated compensation proportional to measured data quality; and governance decisions are executed by DAOs whose voting weights reflect stake-weighted community consensus. The theoretical foundations reviewed in this section undergird the more applied architectural analysis developed in subsequent sections.

16.3. LITERATURE REVIEW

16.3.1 Blockchain for AI Governance

The application of blockchain to AI governance has emerged as a coherent research strand. It focuses on establishing explainability, accountability, and traceability for AI systems deployed in consequential domains. Characterized by the inadequacy of existing accountability mechanisms for automated decision systems. They identified the need for auditable records of model versioning, training data lineage, and inference provenance. Subsequent work demonstrated that blockchain-based model registries can address these requirements. In these registries, model architectures, training datasets, and evaluation metrics are recorded immutably. This approach can satisfy GDPR audit requirements while enabling independent third-party verification of claimed model properties. Traceability has emerged as a particularly active sub-domain. Research proposed a blockchain-based provenance tracking system for machine learning pipelines. The system enables end-to-end auditability from raw data collection through model inference. Their experimental evaluation demonstrated that on-chain provenance records reduced dispute resolution time by 78% in simulated deployment scenarios. More recently, extended this approach to healthcare AI. They showed that blockchain-mediated audit trails for clinical decision support systems could satisfy both regulatory requirements and institutional accountability norms. This was achieved without compromising inference performance. Researchers have also explored explainability mechanisms in conjunction with blockchain. Authors proposed recording explanation artifacts on-chain alongside model predictions. These artifacts include counterfactual exemplars, feature importance scores, and decision boundary visualizations. This approach creates tamper-resistant explanation trails. Although computational overhead remains a concern, their results indicate that selectively anchoring explanation hashes on-chain can achieve the desired audit objectives at an acceptable cost. Collectively, these governance applications establish blockchain as a viable mechanism for implementing the accountability standards required by emerging AI regulatory frameworks, including the EU AI Act.

16.3.2 Blockchain-Enabled Federated Learning

Federated Learning (FL) was originally designed to preserve data privacy by avoiding raw data centralization. However, its reliance on a trusted central coordinator remains a significant architectural vulnerability. A malicious or compromised coordinator can manipulate model aggregation, free-ride on participant contributions, or selectively exclude participants. Blockchain-enabled FL architectures address this limitation by replacing the central coordinator with smart contract logic. This approach provides immutable aggregation records, decentralized consensus on model quality, and transparent incentive distribution. It was introduced by reputation-based federated learning framework. In this framework, participant trustworthiness scores are maintained on-chain and used to weight model contributions during

aggregation. Their system demonstrated superior resistance to model poisoning attacks compared to conventional FL baselines. Contributions from malicious participants were effectively neutralized through reputation-adjusted weighting. The blockchain record of contribution histories also enabled ex-post auditing of model quality drift attributable to specific participants. This capability is absent from conventional FL systems. Incentive mechanism design for blockchain FL has received substantial attention. The formulated participant incentivization as a contract theory problem. They designed payment schedules that encourage truthful reporting of data quality and computational costs. Their smart-contract implementation demonstrated that appropriately designed incentives could sustain high-quality FL participation. This was achieved without relying on altruism or organizational mandates. In contrast, employed Stackelberg game-theoretic models to characterize equilibrium contribution behavior under token-based incentive systems. Their findings indicate that token reward volatility constitutes a primary threat to sustained participation. Privacy preservation within blockchain FL systems requires careful architectural attention. Naïve implementations that record raw gradients on-chain expose sensitive information through gradient inversion attacks. To achieve meaningful privacy guarantees, secure aggregation protocols based on homomorphic encryption and differential privacy mechanisms must be integrated with blockchain coordination. Recent work demonstrated a fully integrated system. It combines zk-SNARK proofs of gradient correctness with differential privacy noise injection. This approach achieves both verifiability and privacy without mutual exclusion.

16.3.3 Decentralized AI Platforms

The practical landscape of decentralized AI is populated by several platforms with varying levels of maturity, architectural orientation, and governance philosophy. SingularityNET, provides a decentralized marketplace for AI algorithms and services. It enables developers to publish, monetize, and compose AI agents through a token-mediated transaction layer. Its AGI token serves as both a utility and governance currency. The platform has facilitated thousands of AI service registrations and is pursuing integration with Cardano's blockchain to improve scalability. SingularityNET's primary architectural contribution is its AI service registry. This smart-contract-based directory enables the creation of composable AI agent networks.

Ocean Protocol is designed to support decentralized data marketplaces. It provides mechanisms for dataset tokenization. It allows AI models to be trained on encrypted datasets without transferring the data from the provider's infrastructure. This approach offers a practical balance between privacy and scalability. It is well suited to sensitive sectors, such as healthcare and financial services, where regulatory requirements restrict raw data sharing but still permit collaborative model development. Fetch.ai adopts a vision centered on autonomous software agents. These agents can perform complex economic negotiations and collaborative tasks without human intervention. The platform's Open Economic Framework (OEF) provides a decentralized layer for agent-based services. The FET token incentivizes participants to contribute infrastructure and network resources. Authors introduces an innovative incentive mechanism known as the Yuma consensus protocol. Under this protocol, TAO tokens are allocated to ML models according to the prediction quality assessed by their peers. The network establishes a competitive marketplace for machine intelligence. A major milestone occurred in 2024 with the formation of the Artificial Superintelligence Alliance through the merger of

SingularityNET, Fetch.ai, and Ocean Protocol. This initiative represents the significant consolidation effort to date and seeks to integrate complementary technologies within a unified decentralized AI framework.

16.3.4 AI Data Marketplaces

Data monetization has emerged as a major application of blockchain-AI integration. It addresses the long-standing mismatch between the value created by data producers and the compensation they receive in centralized data markets. Traditional data brokerage platforms often lack transparency. They provide limited guarantees of data provenance. They also consistently undercompensate the original data providers Blockchain based data marketplaces solve these limitations. They include cryptographic provenance records, licensing agreements enforced by smart contracts and token-based compensation mechanisms. Data provenance is a basic requirement to build trust in decentralized data marketplaces. It provides the auditable records of data provenance, proposed a blockchain-based provenance framework for medical imaging datasets. Their approach enables users to verify the composition of the dataset and how it was collected before they buy it. That mitigates the usual adverse selection problems of centralized data markets. We need to look beyond traditional anonymization for privacy protection. Blockchain-enabled data marketplace architectures leverage techniques. Each technique has its own trade-offs of performance, and scalability. These trade-offs ultimately influence the suitability of each approach for different application scenarios.

16.4. BLOCKCHAIN–AI INTEGRATION FRAMEWORKS

16.4.1 Layered Architecture for Blockchain–AI Systems

Comprehensive Blockchain–AI integration requires a structured multi-layer architecture that cleanly separates concerns while enabling coherent inter-layer coordination. Based on synthesis of existing proposals and analysis of deployed platforms, we identify five essential layers whose interaction defines the functional and security properties of integrated systems.

Layer 1: Data Layer

The Data Layer constitutes the foundation of the integrated stack, responsible for data collection, validation, quality assessment, and provenance management. Data ingestion pipelines aggregate information from heterogeneous sources—sensors, application logs, user-generated content, scientific instruments—and subject it to automated quality screening processes. Cryptographic hashing of raw datasets at ingestion time establishes tamper-evident provenance anchors that propagate through all subsequent processing stages. Critical functions at this layer include: (a) format standardization and schema validation to ensure inter-system compatibility; (b) automated data quality scoring using statistical and ML-based assessment algorithms; (c) privacy screening to detect and redact personally identifiable information before dataset publication; (d) differential privacy noise injection calibrated to dataset sensitivity classifications; and (e) provenance record generation linking each dataset to its collection methodology, collection agent identity, timestamp, and quality scores. These provenance records are anchored to the Blockchain Layer through cryptographic hash commitments, enabling subsequent verification without requiring full dataset access.

Layer 2: Blockchain Coordination Layer

The Blockchain Coordination Layer implements the trust infrastructure that distinguishes decentralized from centralized AI architectures. Smart contract modules govern identity management, access control, consensus participation, payment settlement, and governance execution. Identity management contracts maintain verifiable credentials for all marketplace participants—data providers, model developers, compute providers, and consumers—using self-sovereign identity (SSI) standards that enable selective disclosure of identity attributes without revealing unnecessary personal information. Access control contracts implement programmable licensing agreements that automatically enforce usage restrictions—prohibiting certain downstream applications, enforcing geographic restrictions, or limiting inference call volumes—without requiring active intervention by rights holders. Consensus mechanisms at this layer govern model aggregation in federated learning scenarios, ensuring that aggregation computations produce verifiable results consistent with stated methodologies. Payment settlement contracts automate royalty distribution, usage fees, and contribution rewards in real-time upon fulfillment of verifiable conditions, eliminating the payment delays and disputes characteristic of centralized platforms.

Layer 3: AI/Model Layer

The AI Layer implements model training, inference, optimization, and explainability functions. In a traditional centralized architecture, this layer exists within a single organizational infrastructure. The AI Layer is distributed across multiple participant nodes. The Layer 2 smart contracts coordinate these nodes. Model training is done through federated protocols. These protocols collect local gradient updates without centralizing training data. Model versioning is persisted on-chain. Each model checkpoint is accompanied by immutable records of training dataset compositions. Explainability mechanisms are incorporated in this layer. They satisfy regulatory standards and marketplace confidence needs. The explanation artifacts are generated. These artifacts are stored on the Blockchain Layer to prevent tampering. This allows for auditing of AI decisions after the fact. It does so without requiring access to proprietary model parameters. Model optimization processes can be offered as marketplace services. The optimization results are stored on-chain to reproduce.

Layer 4: Marketplace Layer

The Marketplace Layer provides the economic coordination infrastructure enabling exchange of datasets, trained models, algorithmic services, and compute resources. Four distinct marketplace sub-types serve different participant needs: (a) Data Marketplaces facilitate dataset discovery, quality verification, licensing negotiation, and secure data access; (b) Model Marketplaces enable AI model publication, evaluation, licensing, and deployment; (c) Compute Marketplaces aggregate decentralized GPU and CPU resources into a competitive market for training and inference capacity; and (d) Agent Marketplaces support the discovery and orchestration of autonomous AI agents capable of executing complex multi-step tasks. Pricing mechanisms at this layer range from fixed-rate schedules to dynamic auction mechanisms that respond to supply and demand fluctuations in real-time. Reputation systems aggregate historical transaction records, quality assessments, and dispute outcomes to generate composite trust scores for marketplace participants. Smart-contract-governed escrow

mechanisms ensure that payment commitments are honored before service delivery, while cryptographic proofs of service completion trigger automatic payment release.

Layer 5: Governance Layer

The Governance Layer implements the decision-making mechanisms that enable ecosystem participants to collectively manage protocol upgrades, parameter adjustments, dispute resolution, and compliance monitoring. Decentralized Autonomous Organizations (DAOs) serve as the primary governance vehicle, enabling token-weighted voting on protocol proposals without requiring centralized administrative authority. Governance token holders delegate voting rights, participate in proposal deliberation, and ratify decisions whose execution is automatically implemented by smart contracts. Compliance monitoring mechanisms at this layer interface with external regulatory frameworks, implementing automated checks for GDPR compliance, Know Your Customer (KYC) requirements, and anti-money-laundering (AML) obligations. Reputation systems that aggregate on-chain behavioral records—dispute histories, quality assessment outcomes, participation reliability—provide governance participants with the information needed to make informed voting decisions and trading partner selections.

16.4.2 Smart Contract Integration Patterns

Smart contracts form the programmatic backbone of Blockchain-AI integration. They automate transactional and governance processes that would otherwise depend on trusted intermediaries. Several smart contract integration patterns have emerged as particularly important in decentralized AI ecosystems.

Automated payment contracts release token-denominated compensation after the cryptographic verification of service delivery. This mechanism eliminates the payment delays and disputes commonly associated with manual settlement processes. Licensing agreement contracts encode usage restrictions and royalty schedules as executable logic. Consequently, intellectual property rights and licensing policies can be enforced automatically across unlimited downstream deployments. Resource allocation contracts capture dynamic market mechanisms. These mechanisms allocate scarce computational resources. Revenue-sharing contracts allocate marketplace fees to ecosystem participants. The distribution is based on stake-weighted formulas that align the long-term incentives of participants with the health and sustainability of the ecosystem. Smart contract security and correctness are critical. Developers use formal verification and automated security analysis tools to improve contract reliability before deployment. These tools can be used to verify properties of the contract and to help verify that smart contracts behave as intended under execution scenarios.

Table – 2: Different Tools

Tool	**Classification**	**Execution Style**	**Primary Input**
Slither	Static Analysis	No execution (Code scanning)	Solidity Source Code
Echidna	Dynamic Analysis	Active execution (Fuzzing)	Invariants + Random Transactions

Certora Prover	Hybrid/Formal Verification	Mathematical modelling	CVL Specifications + Bytecode

16.4.3 Tokenized Incentive Models

Token economic design constitutes a foundational challenge in decentralized AI ecosystems, requiring careful calibration of incentives to sustain high-quality participation across all marketplace roles. Utility tokens—providing access to platform services—create demand-side pressure proportional to platform adoption, but suffer from speculative price volatility that can disrupt rational participation incentives. Governance tokens confer voting rights proportional to holdings, creating potential plutocratic tendencies that require mitigation through quadratic voting or conviction voting mechanisms. Staking mechanisms—requiring participants to lock token deposits as performance bonds—provide credible commitment to service quality and create economic costs for malicious behavior. Slashing conditions, implemented as smart contract logic, automatically forfeit staked deposits upon detection of verified protocol violations, providing powerful deterrents against model poisoning, data falsification, and other adversarial behaviors. Reward mechanisms must be designed to compensate not only successful service delivery but also the positive externalities of knowledge sharing, open-source contribution, and governance participation that contribute to ecosystem-level value without being captured in bilateral transactions.

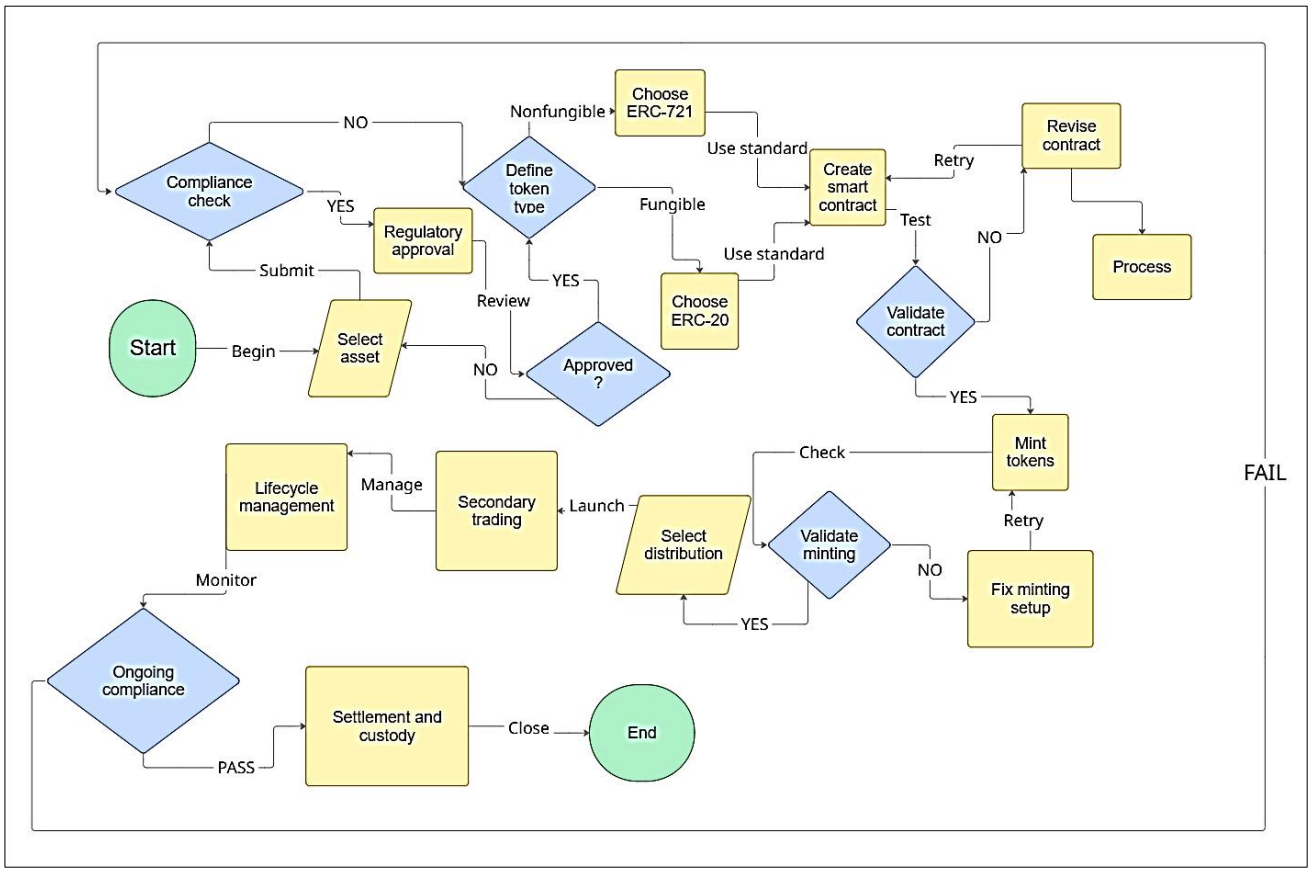


Figure 1: Process Flow of Blockchain Token Creation, Validation, Minting, and Lifecycle Management

16.4.4 Blockchain-Enabled Federated Learning Architecture

The integration of blockchain coordination with federated learning produces an architecture that addresses the central coordinator vulnerability of conventional FL while introducing new challenges of its own. In the blockchain-enabled FL architecture, local model training proceeds as in conventional FL—participants train on local data and compute gradient updates or model delta transmissions. However, the aggregation step is governed by a smart contract that collects participant-submitted model updates, verifies their formal properties (gradient norm bounds, privacy budget compliance), and executes the aggregation algorithm in a verifiable manner. Model aggregation verification can be implemented through several mechanisms: (a) deterministic aggregation executed by all validator nodes with cryptographic commitment to the result; (b) zero-knowledge proof systems that verify aggregation correctness without revealing individual contributions; or (c) trusted execution environment (TEE) attestation that provides hardware-guaranteed privacy for the aggregation computation. Reward distribution is based on verifiable contribution measurement. Participants receive token rewards in proportion to the extent that their local data and computational resources improve model quality. This mechanism allocates greater rewards to contributors who provide higher-value data and computational resources. It also incentivizes high-quality data curation and reliable participation.

16.5. DECENTRALIZED AI MARKETPLACES

16.5.1 Concept and Evolution

Decentralized AI marketplaces provide the institutional infrastructure for exchanging AI-related assets. Unlike conventional platforms that rely on centralized operators, these marketplaces use blockchain-based settlement and governance mechanisms to coordinate transactions and ecosystem management. The decentralized approach is a significant departure. It redistributes economic value, model developers and infrastructure contributors. The participants play different roles. This is supported by a specific incentive structure. Data providers provide training data sets and bear the costs of data collection and other costs. They are paid royalties for the use of their data sets. Model developers invest money on designing algorithms, training them and innovating architecture. Revenue is derived primarily from licensing fees from deployers. Compute providers offer GPU and CPU resources for training and running models. They get paid for how much, how good, and how reliable the computational resources they provide are. Consumers can pay fees to access datasets on demand. Transactions keep the marketplace running and ensure its integrity for the validators. They can earn protocol rewards. These roles are complementary and together they establish an incentive structure. It encourages transparency, decentralization and sustainable growth of the ecosystem.

16.5.2 Marketplace Architecture Typology

Model marketplaces are platforms for evaluating trained AI models. “Model checkpoints are released by developers with verifiable evaluation metrics, architecture specifications and comprehensive usage documentation. Inference interfaces enable prospective licensees to evaluate model performance. The interfaces are used to run test queries without leaking the

model weights. This approach enables informed evaluation before licensing commitments are made. Smart contracts enforce the licensing conditions. These conditions include restrictions on the use of output. The model watermarking techniques enable rights holders to detect and demonstrate unauthorized model deployments. Training datasets are sensitive, thus data marketplaces need advanced privacy mechanisms. Ocean Protocol's compute-to-data paradigm is one of the most advanced in the field. Instead of transferring datasets to model developers, we transfer training tasks to the data provider infrastructure. The tasks are carried out in privacy-preserving computation environments. The raw data stays under the control of the provider and only the trained model parameters are returned. This architecture protects the privacy of data. It helps comply with data residency regulations. It also enables collaborative development of AI models on sensitive datasets that cannot be legally shared via traditional centralized platforms. Compute marketplaces aggregate decentralized GPU resources into competitive markets for training and inference of AI. Resource providers include crypto miners moving away from PoW (Proof-of-Work) consensus mechanisms, academic computing centers with spare computational power and dedicated infrastructure operators. These providers compete on quantifiable attributes such as price, latency and service reliability. Such attributes can be attested by cryptographic performance attestation. More and more compute marketplaces are adopting DePIN (Decentralized Physical Infrastructure Network) frameworks. These frameworks build token-incentivized hardware provisioning networks. They also offer on-chain enforcement of service level agreements. This, in turn, enhances transparency, accountability, and resource utilization across distributed AI infrastructure.

16.5.3 Economic Models

Pay-per-use pricing is a business models in decentralized AI marketplaces. Under this approach the consumer is charged on the basis of actual consumption of the resource. This consumption might be the number of inference API calls, compute hours used, or the volume of dataset access. Smart contracts facilitate usage metering and payment settlement. This reduces the administrative overhead and eliminates the need for trusted middlemen. This model reduces the entry barrier for consumers. It links provider revenue to the value of services provided. But its effectiveness depends on accurate, tamper-resistant metering mechanisms. Such mechanisms need to prevent fraudulent or manipulated reporting of usage. Another pricing model is subscription pricing. The model provides service providers with a predictable revenue stream and consumers with a stable budget. Resources in subscription models can be used if the actual usage differs from the assumptions of the subscription plan. This mismatch leads to under-utilization or to resource shortages. Consumers must lock up a certain number of tokens as collateral to gain access to the services of a marketplace. The level of stake dictates the level of access, or guarantees of quality of service that a user receives. This mechanism aligns incentives throughout the ecosystem. Participants with larger stakes have stronger economic incentives to use services responsibly and to preserve the integrity of the network. The locked capital makes available financial resources to service providers and protocols. These resources can help develop and grow the infrastructure and the ecosystem for the longer term. Royalty mechanisms ensure that data providers and AI model developers are still paid when their assets create downstream value. Royalties spread rewards across the lifetime of deployed models and datasets, rather than restricting rewards to a single transaction. This approach recognizes the continuing contribution of foundational AI assets. It also ensures that creators share in the

economic value generated by subsequent applications, deployments, and derivative services. Such incentive structures encourage sustained innovation. They improve the availability of high-quality datasets and models. They also promote the long-term sustainability of decentralized AI marketplaces.

16.5.4 Trust and Reputation Systems

Trust infrastructure is a critical component of decentralized AI marketplaces. Participants must evaluate counterparties without relying on centralized reputation authorities. To address this challenge, on-chain reputation systems aggregate verifiable records of marketplace activity. These records include transaction histories, successful service deliveries, dispute resolution outcomes, and quality assessment results. The aggregated records are combined into composite trust scores. These scores guide participant selection and marketplace interactions. Since blockchain records are immutable, negative performance histories cannot be selectively removed or altered. This property prevents reputation laundering. It also creates strong incentives for consistent, high-quality service provision. Quality verification extends beyond confirming that a transaction has been completed. It also evaluates the utility and performance of the delivered AI service. For AI models, independent evaluation on held-out test datasets provides an objective measure of model quality. When these evaluations are combined with verifiable computation protocols, they produce cryptographically verifiable evidence. This evidence demonstrates that the reported performance metrics accurately reflect actual model behavior. Such mechanisms strengthen marketplace transparency. They improve participant confidence. They also support informed decision-making during model selection and licensing. For datasets, automated statistical analysis combined with smart-contract-governed validation sample auditing can detect quality inflation claims. Dispute arbitration mechanisms combining on-chain evidence with decentralized juror selection (as implemented in Kleros protocol) provide governance mechanisms for contested quality assessments.

Table 3. Comparative Analysis of Major Decentralized AI Platforms

Platform	Governance	Scalability	Privacy	Token Model	Marketplace Type	Key Limitation
SingularityNET	DAO (AGI token)	Moderate (Ethereum/Cardano)	Basic encryption	Utility + Governance	AI Services & Agents	Limited compute marketplace
Ocean Protocol	DAO (OCEAN token)	Moderate–High	Compute-to-data	Utility + Data NFTs	Data + Compute	Complex data pricing
Fetch.ai	DAO (FET token)	High (Cosmos SDK)	Standard encryption	Utility + Staking	Agent + Data + Compute	Nascent ecosystem maturity
Bittensor	Meritocratic (TAO)	Moderate	Gradient submission only	Emission (Yuma consensus)	Model Intelligence	Sybil vulnerability
ASI Alliance	Unified DAO (ASI token)	Developing	Cross-platform	Merged utility	Full-stack AI	Integration complexity

16.5.5 Comparative Analysis of Existing Platforms

The comparative analysis reveals that no existing platform simultaneously achieves high scalability, strong privacy guarantees, mature governance mechanisms, and comprehensive marketplace functionality. SingularityNET offers the most mature AI services marketplace. However, it lacks native compute provisioning. Ocean Protocol's compute-to-data paradigm provides the strongest privacy architecture. However, it imposes substantial developer complexity. Bittensor's intelligence-as-a-commodity model is theoretically elegant. However, it must address Sybil attack vulnerabilities before it can achieve broad enterprise adoption. The emerging ASI Alliance represents an effort to provide comprehensive capabilities by combining complementary decentralized AI technologies. However, the integration of multiple platforms introduces significant technical complexity and poses near-term implementation challenges.

16.6. CHALLENGES AND LIMITATIONS

16.6.1 Technical Challenges

Scalability remains the paramount technical challenge for blockchain-integrated AI systems. Today’s layer 1 blockchains suffer from scalability bottlenecks. Ethereum can handle a limited number of transactions per second under normal operating conditions. This is not enough throughputs for the high-frequency micro transaction workloads for AI marketplaces. Payments, resource allocations and service requests are ongoing in these marketplaces. Effective throughput is greatly improved by layer 2 scaling solutions such as optimistic rollups, zk-rollups, and state channels. They do this by pushing much of the transaction processing off the main blockchain. These approaches come at the cost of increased architectural complexity, liquidity fragmentation, and new security assumptions. These problems have to be dealt with carefully. Another big challenge is the computational expense of training AI. Training large AI models can take days or even weeks of nonstop computation on GPUs. Resource-consuming workloads are different from the millisecond-scale execution model for which smart contracts were built. AI computation cannot be done on-chain. That it needs instead are hybrid architectures that coordinate off-chain computation with on-chain verification and settlement. Another challenge is the interoperability across heterogeneous blockchains ecosystems. AI asset owners willing to use services on different blockchain platforms have to use different token standards, consensus mechanisms, and smart contract environments. Partially solving this problem are cross-chain bridge protocols that allow transferring assets. These bridges have been proven to have significant security issues. Cross-chain bridge exploits cost more than USD 2 billion in 2022 alone. This result highlights the security issues of blockchain interoperability. Storage limitations prevent the direct use of blockchain in AI applications. Weights of AI models, large training datasets, and detailed provenance records are several orders of magnitude larger than the on-chain storage capacity of current blockchain networks. Thus, hybrid storage architectures are used in practical blockchain–AI systems. In these architectures, cryptographic hashes and metadata are stored on-chain to provide integrity and provenance. The underlying data is stored in decentralized off-chain storage such as IPFS, Filecoin or Arweave. This strategy leverages blockchain’s immutability and scalability needed for storing big AI assets.

16.6.2 Security Challenges

Model poisoning attacks are a serious threat to blockchain-enabled federated learning. In these attacks, malicious participants send malicious gradient updates. The updates are designed to degrade the performance of the aggregate model or re-embed backdoor vulnerabilities. Partial mitigation is provided by Byzantine fault-tolerant aggregation algorithms (Blanchard et al., 2017). However, the effectiveness of these algorithms decreases when the fraction of malicious participants approaches the theoretical threshold that these algorithms can tolerate. The blockchain must be transparent to be auditable. At the same time, it gives adversaries complete visibility into the patterns of participation. This visibility may enable attackers to infer system behavior and develop more targeted attack strategies. This visibility enables more sophisticated coordinated attacks than are feasible against centralized systems.

Smart contract vulnerabilities have resulted in cumulative losses exceeding USD 5 billion across the blockchain ecosystem. These vulnerabilities include re-entrancy attacks, integer overflows, access control failures, and oracle manipulation. AI marketplace contracts managing substantial token reserves are attractive targets for adversarial exploitation. Sybil attacks pose a significant threat. Adversaries create multiple pseudonymous identities to manipulate reputation systems, governance votes, or incentive distributions In these attacks. The threat is severe in permissionless ecosystems, where identity verification is constrained. Adversarial AI attacks introduce additional risks. These include prompt injection, jailbreaking, and model extraction, all of which are explicit to AI service marketplaces.

16.6.3 Ethical Challenges

Algorithmic bias embedded in training data can be amplified and entrenched by marketplace mechanisms. These mechanisms reward high performance on biased evaluation benchmarks. Decentralized governance structures may worsen rather than reduce bias concerns. This is especially likely when token holdings are concentrated among demographically homogeneous groups whose interests differ from those of affected communities. Explainability requirements are increasingly mandated by regulation. However, they are structurally at odds with the black-box nature of high-performance deep learning models and the privacy constraints of federated learning systems. Accountability attribution in decentralized systems is inherently complex. When harmful AI decisions result from models trained on marketplace datasets using distributed computing resources, identifying responsible parties for legal and regulatory purposes becomes a novel challenge.

16.6.4 Regulatory Challenges

The GDPR requires that personal data processing be grounded in a specific lawful basis. It prohibits cross-border data transfers without adequate safeguards. These requirements create challenges for decentralized AI marketplaces operating across national jurisdictions. The immutability of blockchain records appears to conflict directly with the GDPR's right to erasure. The technical solutions provide partial mitigation. The pseudonymous identity model characteristic of permissionless blockchains may be incompatible with Know Your Customer (KYC) and Anti-Money Laundering (AML) obligations. These obligations apply to financial marketplace operations in many jurisdictions. Regulatory fragmentation across jurisdictions

further complicates compliance. The EU AI Act, the US executive orders on AI, China's generative AI regulations, and emerging national frameworks in India, Brazil, and Singapore all impose different requirements. As a result, globally operating decentralized platforms face substantial compliance complexity. Compliance with one jurisdiction's requirements may conflict with those of another. These conflicts are particularly evident in areas such as data localization, mandatory human oversight, and technical transparency obligations. The decentralized governance structure of DAO-operated platforms creates additional uncertainty regarding which legal entity bears regulatory accountability.

16.6.5 Economic Challenges

Token price volatility creates persistent instability in marketplace economics. When service prices are denominated in volatile tokens, real-price uncertainty discourages long-term planning by both providers and consumers. This uncertainty also reduces investment in high-quality data curation and model development. Incentive misalignment can emerge when participant objectives diverge from the long-term health of the ecosystem. Contributors may prioritize short-term token extraction over sustained quality provision. Validators may favor transaction throughput instead of rigorous quality verification. Governance token holders may support parameters that maximize short-term token prices rather than long-term ecosystem value. Market manipulation introduces an additional dimension of economic risk. Coordinated trading, governance attacks, and wash trading in synthetic reputation systems can all undermine marketplace stability.

16.7. PROPOSED TDIMF

16.7.1 Framework Overview

The Trusted Decentralized Intelligence Marketplace Framework (TDIMF) is an original conceptual architecture. It synthesizes the analytical findings of this review into a coherent system design. TDIMF addresses the critical integration gaps identified in the literature. These gaps include the absence of unified frameworks that coherently address architecture, economics, governance, and security as mutually constitutive dimensions. The framework addresses these challenges by specifying their interactions through five interdependent layers with well-defined interfaces and coordination protocols. The TDIMF is distinguished from existing frameworks by three core design principles. The first is Privacy-by-Architecture. Under this principle, privacy preservation is a structural property of the system design rather than a compliance overlay. The second is Incentive Coherence. This principle ensures that economic incentive mechanisms are designed for dynamic stability under adversarial conditions rather than for static equilibrium analysis. The third is Governance Legitimacy. It ensures that decision-making processes are designed to achieve outcomes that are considered legitimate by all ecosystem stakeholders rather than merely being technically correct according to voting rules.

16.8.2 TDIMF Layer Specifications

Layer 1: Data Trust Layer

The Data Trust Layer implements privacy-by-default data handling through differential privacy noise injection, federated data validation, and cryptographic provenance anchoring. Every dataset entering the marketplace is assigned a Privacy Risk Score. This score is computed through automated PII detection, re-identification risk analysis, and sensitivity classification. Datasets that exceed the threshold risk score undergo mandatory privacy-enhancing transformations before they are admitted to the marketplace. Provenance records are anchored to Layer 2 through zk-SNARK commitments. These commitments verify data properties without revealing the underlying data contents.

Layer 2: Blockchain Coordination Layer

The Blockchain Coordination Layer implements a modular smart contract architecture. This architecture enables selective upgradeability without requiring full protocol migration. Core identity, payment, and provenance contracts are deployed as immutable singleton instances. These instances are verified through formal methods. Auxiliary contracts implement marketplace logistics, quality verification, and governance voting. They are deployed as upgradeable proxies and are subject to DAO-governed upgrade processes with mandatory time delays. The coordination layer also interfaces with multiple blockchain networks through standardized cross-chain messaging protocols. This approach enables asset portability without relying on vulnerable bridge architectures.

Layer 3: Federated Learning Layer

The Federated Learning Layer implements the TDIMF's core intelligence infrastructure through a smart-contract-coordinated federated training protocol. Local training proceeds using standardized federated learning APIs compatible with TensorFlow Federated, PySyft, and Flower frameworks. Gradient submissions are validated through lightweight zk-SNARK circuits. These circuits verify gradient norm bounds and differential privacy budget compliance before the coordination contract accepts the submissions. Model aggregation uses robust aggregation algorithms, including Multi-Krum and FedAvg with Byzantine clipping. Validator nodes execute these algorithms deterministically. Contribution measurement employs Shapley value approximations computed through structured sampling protocols. This approach provides a principled incentive allocation mechanism that rewards each participant's marginal contribution to model quality.

Layer 4: AI Marketplace Layer

The AI Marketplace Layer implements integrated sub-marketplaces for data, models, compute, and agents within a shared economic framework. Cross-marketplace composability enables complex service compositions. These include purchasing training data, executing a training job, publishing the resulting model, and deploying it for inference. All of these operations can be completed within single atomic transaction bundles governed by multi-step smart contract workflows. Dynamic pricing mechanisms adapt to real-time supply and demand conditions through on-chain oracle feeds. These feeds aggregate peer-submitted price observations. Model evaluation-as-a-service enables prospective licensees to assess model quality through

standardized benchmark evaluations executed in trusted execution environments. The evaluation results are recorded on-chain for future reference.

Layer 5: Governance Layer

The Governance Layer implements a multi-level governance architecture. It distinguishes among protocol-level decisions, marketplace-level decisions, and operational decisions. Protocol-level decisions require supermajority consensus among long-term stakeholders. Marketplace-level decisions require a simple majority among active participants. Operational decisions are delegated to elected councils. Conviction voting mechanisms allow voting weight to accumulate over time in proportion to token commitment duration. This approach mitigates the short-term price speculation that distorts conventional token-weighted governance. AI-assisted governance analytics provide participants with objective performance data, proposal impact assessments, and historical precedent analysis. These resources support informed voting. Cross-ecosystem governance coordination mechanisms further enable coherent policy development across the fragmented decentralized AI landscape.

Table 4. TDIMF Comparative Advantages vs. Existing Frameworks

Dimension	Existing Approaches	TDIMF Innovation
Privacy	Peripheral compliance overlays	Privacy-by-architecture with zk-SNARK provenance
Incentive Design	Static equilibrium analysis	Dynamic stability under adversarial conditions
Governance	Simple token-weighted voting	Multi-level conviction voting with AI-assisted analytics
Security	Component-level analysis	System-level compound risk characterization
Interoperability	Single-chain assumptions	Multi-chain coordination via standardized messaging
Scalability	On-chain computation	Hybrid on/off-chain with cryptographic verification
Explainability	Retroactive addition	Co-designed explanation anchoring at Layer 3

16.8. CONCLUSION

This paper has pursued a systematic and integrative analysis of Blockchain–AI convergence. It brings together a critical mass of scholarly literature to describe the state of the art, identify key gaps and propose new frameworks to advance research and practice in this fast-moving domain. The main finding is that blockchain and AI are complementary technologies. Their combination can trust properties that neither technology can attain. The five-layer integration framework developed in Section 4 provides a principled architectural vocabular. It separates the concerns of data provenance, blockchain coordination, AI model management, marketplace economics and governance legitimacy. It describes the dependencies that exist between these layers. Section 5 of the comparative platform analysis clearly shows that there are fundamental trade-offs between scalability, privacy, governance maturity, and marketplace comprehensiveness in existing ecosystems. The proposed TDIMF framework is designed to

overcome these trade-offs through architectural integration rather than component-level optimization. The Trusted Decentralized Intelligence Marketplace Framework (TDIMF) is designed around three core principles. These are Privacy-by-Architecture, Incentive Coherence, and Governance Legitimacy. These principles together surmount the most stubborn restrictions of existing decentralized AI systems. Privacy-by-Architecture embeds privacy protections into the system architecture, rather than considering privacy as an optional feature. Privacy preservation is an inherent property of the framework. This method minimizes the risk of regulatory non-compliance and implementation mistakes. Incentive Coherence guarantees that the economic interests of all participants are aligned over time, even in adversarial conditions and changing market dynamics. This is done by creating reward systems that always reward honest participation and high-quality contributions. The framework enhances the long-term stability of AI ecosystems. Governance Legitimacy Ensures that the protocol's decision-making is representative of the wider ecosystem rather than a small group of people. This principle promotes transparent, accountable and inclusive governance. It enhances stakeholder confidence in decentralized decision-making processes. The framework has practical implications for a number of stakeholders. AI researchers and developers can follow a structured approach to the development of trustworthy, auditable and privacy-preserving training pipelines. These pipelines satisfy regulatory requirements without sacrificing model performance. Data providers receive architectural guidance on how to monetize their datasets while still having meaningful control of the downstream use of the data. Platform architects gain a systematic understanding of the trade-offs. Regulators get a systematic view of the compliance challenges of decentralized AI. They learn about technical mechanisms that can be used to support accountable governance. The eight research directions identified in Section 7 provide a comprehensive agenda for future work. Whether decentralized intelligence marketplaces develop into reliable infrastructure for the global AI economy will depend on progress in these areas. High on the agenda are the development of quantum-resistant cryptographic techniques, the improvement of the efficiency of privacy-preserving computation, and the design of robust and inclusive governance mechanisms. Solving these challenges will require sustained multidisciplinary collaboration among researchers in cryptography, economics, law, distributed systems and artificial intelligence engineering.

CHAPTER 17

APPLICATIONS ACROSS HEALTHCARE, FINANCE, IOT, AND CYBERSECURITY

Contributed by: Dr. Sanjay Nag, Aditya Dutta, Tanmay Saha

17.1. INTRODUCTION

Blockchain Technology as the Most Powerful Invention in the Digital Age. People accept blockchain technology as one of the most revolutionary innovations of modern time. Blockchain technology is understood as an innovative technique due to the fact that the basic principle of this technology is the existence of an open record system. Since this record system provides confidentiality and safety of information handling, blockchain technology is widely used in various business fields. The concept of blockchain technology was introduced by Satoshi Nakamoto, an anonymous inventor, in 2008 in connection with the creation of the first electronic currency Bitcoin. The analysis of blockchain technology proves that it is the cornerstone of Bitcoin. The difference between traditional middle-ranked data storage and blockchain becomes evident. The traditional method of storing data requires the services of intermediaries to manage money transactions. The use of blockchain technology becomes evident. Fast development of digital technologies - Our interconnected world today. There are some professional areas such as medicine, financial management, IoT technologies, and cybersecurity that consider blockchain as a revolutionary technology. Until now, previous systems worked with fragmented data storage and third parties, but blockchain technology offers transparency and trust. For medicine, blockchain technologies help store and exchange medical data. Its capacity allows keeping permissions, tracking medicines, and making insurance payments faster. In the finances sector, blockchain technology improves payment transaction processes. The use of blockchain technology in the sector of connected devices implies that it is easy to monitor devices. Data obtained from sensors is protected and procedures executed based on digital contracts when blockchain technology is used. Blockchain technology solves the issue of tracking goods in hubs. In addition, when it comes to security on the internet, blockchain improves identity management processes. It is also noted that blockchain produces immutable audit logs. Blockchain technology guarantees secure data flow and quick detection of threats. These changes show that blockchain technology is no longer a tool for digital currencies. Instead, blockchain technology becomes a layer for all business areas to send data safely and to make tasks happen by themselves. Experts claim that blockchain technology builds trust between people. This section of writing talks about how blockchain technology is used to fix real problems. Speed and quality are increased and entry to services is

provided because blockchain technology is used in these areas. The hard parts of technology and problems with rules in real-world use of blockchain technology are also explained.

17.2. Applications of Blockchain Across Healthcare

The healthcare industry is currently going through changes thanks to blockchain technology. This technology, which was designed to support cryptocurrency transactions, is evolving into a very important part of the digital infrastructure used to convert old-fashioned and unconnected information systems into connected and secure networks. By means of applying cryptographic verification and immutable records, blockchain technology enables a decrease in the involvement of governmental institutions and intermediaries. This technology resolves some issues in health care: fragmented medical records, unsecured databases, counterfeit medications, and delayed payment processes documented in insurance policies. Additionally, in combination with smart contracts, IoMT, and artificial intelligence, blockchain introduces a novel level of health care practice centered around the needs of patients.

17.2.1. Electronic Health Records (EHR) and Interoperability

The traditional data transfer between hospitals has been very slow and expensive because of the fact that each hospital runs its separate database. This situation creates problems during emergencies requiring quick access to the medical history of a person. However, blockchain technology can help with this issue by enabling peer-to-peer verification of information. In comparison with the traditional system where hospitals must send formal requests for information to the authorities, the hospitals in the blockchain will be able to verify the medical record of their patient in a matter of seconds. The Estonian e-health system has implemented the use of cryptographic hash functions to ensure secure data exchange in real time.

National Digital Health Infrastructure

Blockchain technology also aids in the creation of nationwide healthcare systems. It helps governments link public hospitals, private clinics, and laboratories through decentralized ledgers. The Ayushman Bharat Digital Mission of India is a prime example, as it has already assigned 670 million Ayushman Bharat Health Account IDs. However, the current ABHA technology uses a federated database (which is dispersed across several collaborating machines), it is ideal for future blockchain use. Blockchain will provide smart contracts to facilitate the patient consent management, thus, eliminating duplicate medical tests and making the healthcare system cheaper.

17.2.2 Pharmaceuticals' Logistics and Traceability

In the pharmaceutical industry, logistical activities are intricate and exposing them to fraud. Blockchain technology acts as a public ledger providing manufacturers, logistic providers, and regulatory bodies with credibility.

- ***Paperless and Counterfeit Proof:*** The platform is utilized to replace paper registration and reduce fake medicine penetration in the supply chain.
- ***Fast Regulation:*** Real-time tracing contamination in batches enables prompt targeted recalls.
- ***Tokenization of Healthcare Assets:*** Physical batches of medicines can be tokenized for the local pharmacies to check authenticity of medications by scanning.

- ***International Logistics:*** The blockchain-based solution simplifies the clearance and control of quality of international deliveries.
- ***Automatic Payments:*** Automatic payment for delivery of consignments and notifications in case of the temperature changes in the cold chain of vaccines become available.
- ***Digital Twin:*** Digitalized shipments will enable the automation of logistical operations.

17.2.3. Decentralized Clinical Research and Telemedicine

Blockchain can serve as an important infrastructure that allows decentralized clinical trials and telemedicine to take place. In centralized clinical trials, the biggest challenge in the context of the new technology is related to the role of central data broker or Contract Research Organization (CRO). In decentralized clinical research applications, workflows can be automated, consents can be secured, and rewards can be given to the participants through the use of digital health wallets. As for telemedicine, blockchain creates opportunities for ensuring the integrity of peer-to-peer consultations between the doctors and patients. Although blockchain gives a broad access to clinical trials, it is important to point out that blockchain has certain drawbacks associated with its use in healthcare, including the specificity of laws concerning data protection.

17.2.4. Digital Identity, Consent, and Data Protection

Identity verification in the healthcare sector can often be both repetitive and invasive. Blockchain can resolve such issues with the use of Self-Sovereign Identity (SSI), which allows the individual to take control of their own identity. Thanks to the zero-knowledge proofs (ZKP) technology, the patient can demonstrate his/her compliance with specific medical requirements, such as having insurance, without disclosing sensitive information from the patient's medical file. This technology can support doctors in meeting the new privacy regulations concerning patient data protection. While having benefits in the memory of the access history and record keeping, it faces the challenge of reconciling data accessibility with the patient's right to be forgotten.

17.2.5. Health Data Tokenization and Genomics

Studies of illnesses depend heavily on vast amounts of data related to genomic sequences and medical records of patients. The process of data tokenization makes it possible to obtain ownership of the real-life health data through the process in which the data is stored in the form of a digital token on blockchain. The patients can transfer their data with the help of this technology and make profits from the transaction. The research organizations have started using this approach due to its useful features and a focus on the protection of patients' rights. The healthcare organizations use the tokenized data effectively with the purpose of implementing the necessary access limits and enabling the patients' clinical work to become faster and more secure. The technology serves as a basis for further progress in health studies.

17.2.6. Health Insurance and Claim Processing

Health insurance claims are difficult to process and involve many loopholes. The use of smart contracts helps to eliminate all these problems since they are programs that can verify and approve patients' claims based on hospital's data and tests results. It helps to avoid the use of

TPAs, thus speeding up the discharge process and increasing the efficacy of treatment. One of the key innovations in this field is parametric insurance. By means of blockchain and IoMT technologies (like glucose meter or pacemaker), the insurance company can get all the necessary information about a patient's health status in real time and approve the respective payments if certain criteria were fulfilled.

17.2.7. National Health Registries and Public Health Surveillance

When employed on the macro scale, blockchain technology provides protection for National Health Registries and facilitates issuance of digital health identities by the government. By connecting hospitals from various regions into one unified blockchain network, healthcare agencies achieve a rapid and transparent process of information collection, which is essential for epidemiological monitoring and epidemic surveillance. The implementation of this system on the national scale implies overcoming the challenges concerning cybersecurity, privacy of personal data and internet access by rural populations. In conclusion, these issues facilitate the establishment of fully Integrated HealthTech Ecosystems.

17.2.8. Case Study: Blockchain-Powered Ecosystem for India's ABHA

The Ayushman Bharat Digital Mission (ABDM) is likely among the world's greatest projects concerning digital health. Its hallmark feature is the Ayushman Bharat Health Account (ABHA) — a unique 14-digit ID that enables comprehensive management of patient health records throughout the country. The use of blockchain technology will create a highly secure and patient-friendly decentralized system.

The Challenges

In the past, medical records in India were dispersed among public and private organizations and medical records were in the form of files. This made it hard for people to keep track of their own medical history and records. Since the introduction of digital ABHA ID, there is a possibility of sharing information securely between medical organizations; however, hacking and misuse of the information should be avoided at all cost.

Importance of the ABHA ID

- **Universal Interoperability:** Rural patient can immediately send his medical history to the metropolitan doctor.
- **Continuity of Care:** The full medical history eliminates misdiagnosis and repeat procedures.
- **Public Health Analytics:** Pool of standardized anonymized data helps scientists to track the disease development.

Blockchain Integration into ABHA

The concept of integrating blockchain technology into the ABHA, which is suitable for nationwide application (for example, Hyperledger Fabric), indicates:

- **Decentralized Identifiers (DIDs):** the ABHA ID is based on a blockchain platform, enabling patients' self-sovereignty.

- **Hybrid Data Architecture:** large medical documents (e.g. MRI, X-ray study results) will be stored off the blockchain in hospitals' electronic data banks while their cryptographic hash will be placed on the blockchain to ensure effective operation of the network and adherence to regulatory standards.
- **Smart Contract Approval:** data requests will trigger smart contracts, which will send notifications to patients' devices asking for consent.

Important Benefits of Blockchain Integration

- **Granular Consent:** The patient can give permission for limited access (for 24 hours) to specific documents (such as those concerning heart disease history) and revoke it when the time has run out.
- **Immutable Audit Trail:** The access of every data record is recorded permanently and is impossible to change, which decreases the possibility of medical fraud**.**
- **Zero-Knowledge Proof:** The patient may obtain cryptographic confirmation of their qualification for treatment and they do not have to reveal their full medical record**.**
- **No Single Point of Failure:** Thanks to the government and hospitals authenticated nodes distributed system data is protected from hacking and local power cut-offs.

17.2.8.1. Timeline of Blockchain and ABHA Innovation (2020–2026)

Conceptualization and Pandemic Stimulu - 2020

- **ABHA:** Kick-off of the trial stage of the National Digital Health Mission (NDHM) project, the first mention of the idea of "Health ID."
- **Blockchain**: The global coronavirus pandemic reveals critical vulnerabilities of supply chains. Which triggering the launch of pilot projects of decentralized monitoring of PPE and vaccines.

Official National Deployment - 2021

- **ABHA:** Introduction of the Ayushman Bharat Digital Mission (ABDM) throughout India, establishing the 14-digit ABHA Card.
- **Blockchain**: Hospitals switch from centralized storage systems to distributed cryptographic infrastructure to protect Electronic Health Records (EHR).

Fast Implementation and Ecosystem Development - 2022

- **ABHA:** System grows to more than 150 million IDs, involving healthcare facility and professional registries that digitally connect doctors and patients.
- **Blockchain:** Smart contracts are deployed, automating credential validation and enforcing data sharing rules between hospitals.

Incentive for Digitization and Asset Protection - 2023

- **ABHA:** Digital Health Incentive Scheme (DHIS) proves effective in rewarding hospitals that digitize the linkage between patients' records and their IDs.
- **Blockchain:** Widespread adoption in pharmaceutical track-and-trace provides security of sensitive healthcare assets distribution and prevents fake medicine circulation.

Mass Implementation and Enhanced Privacy Technologies - 2024

- **ABHA:** Number of accounts rises above 670 million; healthcare coverage is expanded to cover all elderly, hence requiring immediate digital verification.
- **Blockchain:** Deployment of Zero-Knowledge Proofs (ZKPs) and Decentralized Identifiers (DIDs) allow patients to prove their health issues without revealing whole health history.

AI and IoMT Integration - 2025

- **ABHA:** The ecosystem matures into an active data exchange platform; anonymized data goes to predictive AI models providing effective public health surveillance.
- **Blockchain:** Successful integration with IoMT allows wearables to send untampered vital signs information to blockchain ledger.

Trustless Interoperability Era - 2026

- **ABHA:** Creates the basic foundation of Indian healthcare, providing instant and paperless insurance claims processing and continuity of care throughout the country.
- **Blockchain:** Functions as the final "trust layer" where smart contracts process claims automatically based on immutable records.

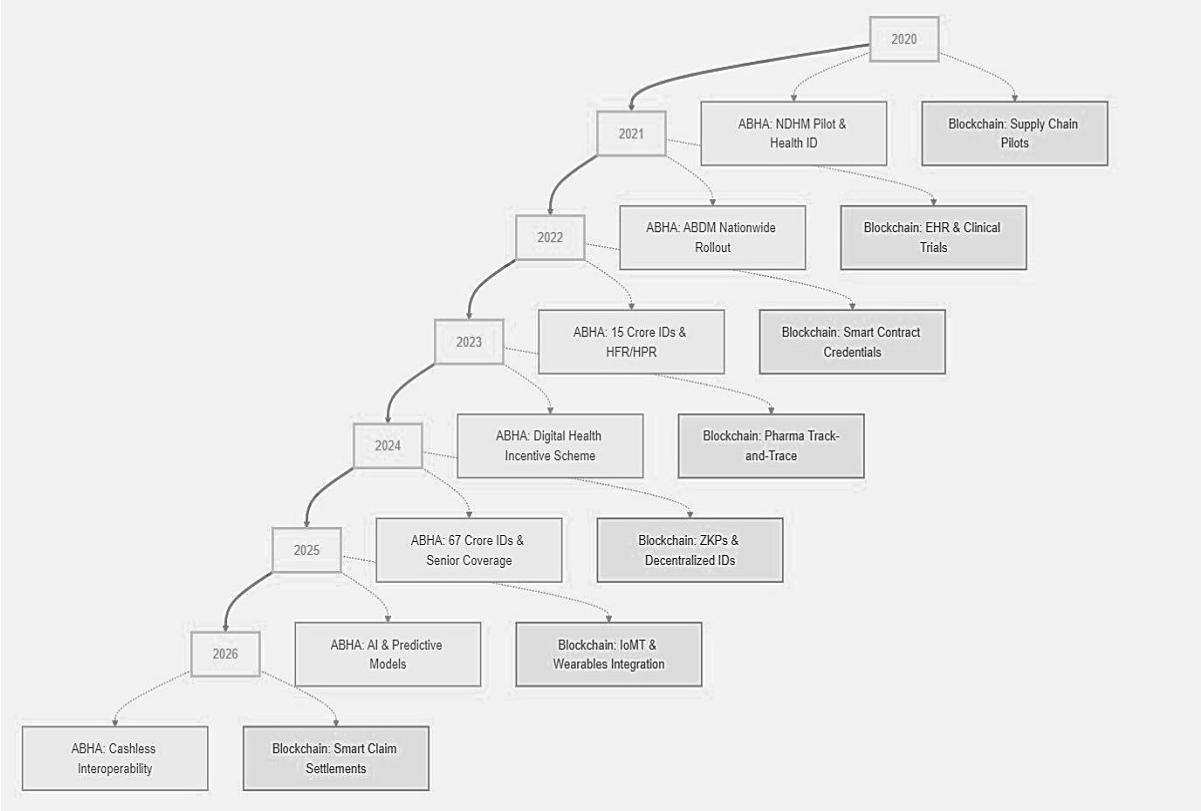
17.3 APPLICATIONS OF BLOCKCHAIN ACROSS FINANCE

Blockchain technology is changing the face of finance, transitioning from trust-based systems to decentralized networks that offer visibility and robustness. Originally developed for making cryptocurrencies, blockchain evolved into an essential mechanism for value transfers based on cryptographic verification, immutable records, and decreased need for middlemen. Blockchain deals with some of the most demanding issues of conventional finance sector such as expensive transactions, inefficient infrastructure, and limited availability of financial products for all possible clients. By combining blockchain technology with smart contracts tokenization tools and even artificial intelligence, new possibilities for FinTech innovations and enhanced financial access are created. The decentralized nature of blockchain technology helps to increase reliability and transparency of financial transactions at the same time. And that is important, because financial opportunities are provided to those who were deprived of them in the past small business companies and individuals in developing economies.

17.3.1. Payments and Settlement Systems

Cross-Border Payments - The utilization of blockchain technology can significantly enhance international payment transfers by removing the need for middlemen, enabling faster processing and increasing transparency. Traditional cross-border payment methods often rely on the intermediary services of correspondent banks that slow the procedure down, drive costs up, and create uncertainty, which is particularly noticeable in remittance payments sent to the developing countries. However, blockchain payment systems allow for direct transactions between users and provide atomic settlement (full or no payment) consequently reducing the time needed for settling the payments from several days down to a few minutes, along with real-

time visibility of the payment status. In addition, these payment systems run 24/7 without delays related to weekends or holidays. The UBS-Ant International collaboration, which was announced in November 2025, is a genuine case in point, as it will entail the introduction of tokenized deposits and UBS Digital Cash to facilitate real-time transfers of several currencies among different organizations without any of the typical limits. Indeed, these are indicative of the way banks and fintech companies are integrating blockchain for most effective and transparent operations. Empirical studies prove that, as they say, costs are lower, settlements happen quicker and audits are easier for regulators.



- **Figure – 1: Chronology of annual ABHA development associated with blockchain.**

Domestic Payment Infrastructure - Blockchain is changing home payment systems, improving efficiency, resilience, and inclusion thanks to DLT. The concept of interoperable blockchain networks allows banks, fintechs, and payment processors to work together more effectively. An instance proving the viability of this technology is India's UPI, making it possible to handle 14+ billion transactions monthly since 2016. While not operating on the blockchain, UPI gives a good example of a supportive framework for new blockchain-enabled technologies like the Unified Markets Interface. The benefits obtained from using blockchain-based domestic payments include faster settlement times (from ~24 hours to under five minutes), lower transaction errors rate (65% or more fewer), and reduced operational costs (20% or more lower).

17.3.2. Trade Finance and Supply Chain Finance

Applications of blockchain in trade finance and supply chain finance

The joint permanent record helps to simplify documentation processes and to enhance the level of transparency by allowing importers, exporters, banks, insurers, and providers of logistics to

reach the same trade records and to review them which reduces typical documentary procedures and cuts fraudulent actions at the same time. Faster and more secure financing decisions together with obtaining verified trade information, results in minimizing information gaps and speeding up the procedures of due diligence which allows enterprises of small and medium businesses to obtain financing decisions. Trade asset tokenization results in the opportunity for fractional ownership through tokenization of receivables and trade-backed assets so the second-round transactions can be done easily. The use of platforms based on public ledgers (such as Linklogis based on XRP ledger) simplifies cross-border settlement and circulation of assets making it faster and less dependent on correspondent banks. Smart contracts automation refers to the operation of smart contracts without human involvement (smart contracts do everything by themselves - processing payments as soon as the goods are verified, coding the conditions of output - releasing collateral for example - and applying trigger financing). Employing artificial intelligence along with data analytics makes it possible for businesses to execute credit assessment and risk management based on verified signals received from the blockchain. The introduction of blockchain ensures the visibility of the supply chain and maximizes the efficiency in managing working capital. Automated financing through the use of digital twins implies that the implementation of blockchain with the digital twins enables the creation of virtual asset models in real-time, so financing can occur when the conditions predetermined by the model are met.

17.3.3. Decentralized Finance (DeFi)

In the area of what is known as Decentralized Finance (DeFi) we see that which traditional financial services like lending, borrowing, trading and asset management are included all of this in a no longer intermediary dominated space. In DeFi instead of that which is provided by central authorities we have smart contracts which bring about automation and reduce the role of banks; also in play is the digital wallet which increases access. The blockchain brings to the table transparency, immutability and decentralized accountability which in turn also present reliability, security and speed. DeFi applications like decentralized lending platforms and decentralized exchanges present how blockchain is put to good use in the financial world. With self-managed lending services which run the flow of loans, collateral management and interest calculation, the process comes to feel more algorithmic than manual. In terms of decentralized exchanges, we see the use of AMMs which enable direct P2P asset swap it is a system in which liquidity is constantly being provided by the network. But achieve scaling of these is easier said than done which we see in the form of smart contract vulnerabilities, issues with regulation, privacy issues and the lack of proper customer protection measures. The latest in DeFi are privacy first compliance solutions which are put forth to combine the benefits of decentralization with the requirements of regulation thus creating new better and safer financial systems.

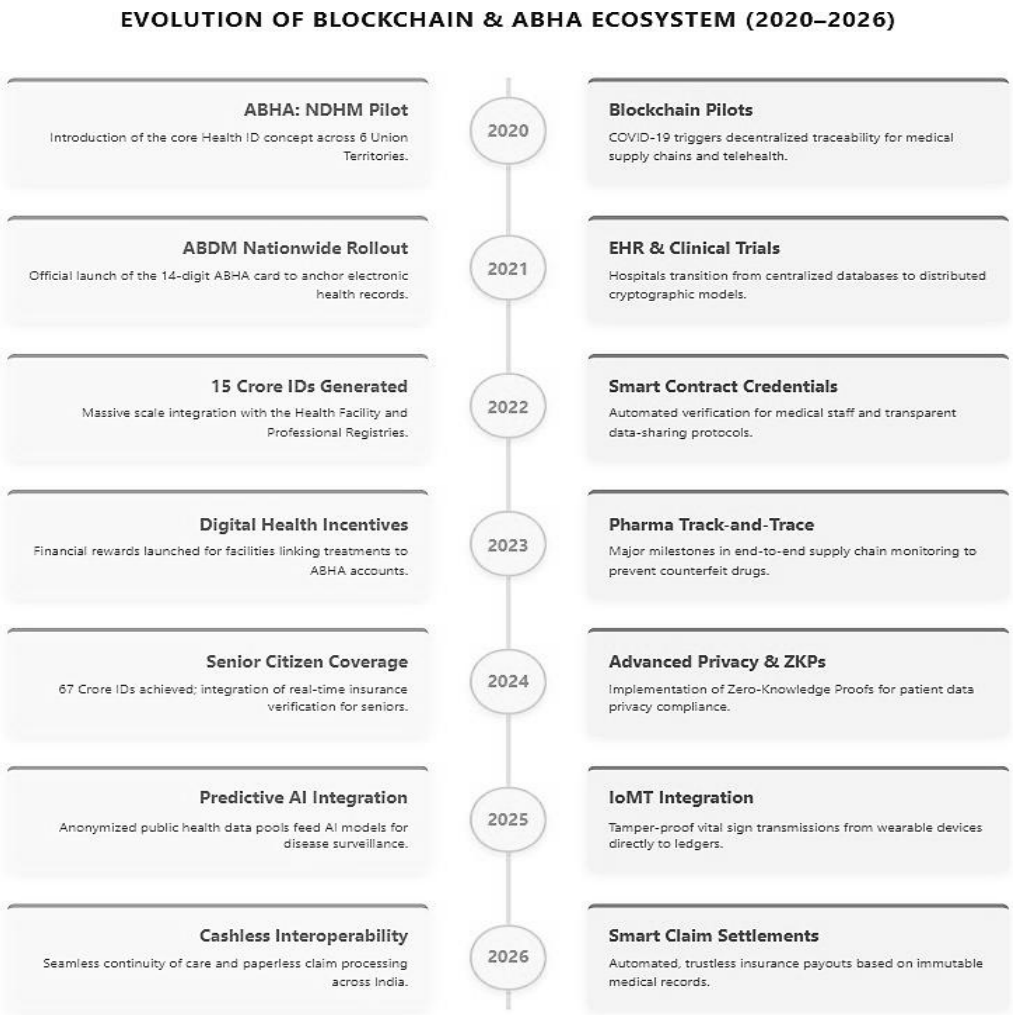


Figure – 2: Chronology of the parallel development of ABHA platform and health blockchain applications.

17.3.4. Digital Identity, KYC, and AML Compliance

Digital finance reports that it is at a crossroads with identity verification which is marred by the issue of cost which in turn causes repetition and what we may term a breach of privacy. Out of this comes Self Sovereign Identity (SSI)based on blockchain which puts power back in the hands of the individual as they put forth their own identity data and only share out what is relevant and which is proven to be true. Also, thanks to the use of blockchain tech, verifiable credentials, and zero knowledge proofs financial institutions are able to do great job at identity validation at the same time exposure of personal info is reduced. Other blockchain-based KYC/AML technologies have also addressed real problems, such as privacy-preserving customer authentication, the tracking of transactions, and regulatory rule observance. For example, RegKYC allows a user to prove that they are abiding by the rules without revealing their personal information, thus enabling financial institutions to meet their security obligations. Much as blockchain provides significant assistance to AML by allowing for the storage of transaction details and the automated execution of smart contracts, it has left some issues unresolved, including those relating to anonymous transactions, the correct interpretation of the terms of regulation, and the setting up of an appropriate balance between privacy and the need for regulatory control over finance.

17.3.5. Asset Tokenization and Securities

The term asset tokenization can be defined as "conversion of ownership rights of real-world assets into tokens on a blockchain platform." The main implication of this process is an atomization of ownership rights, acceleration of transactions, getting transparency due to fast settlements, as well as possible liquidity increase for assets that are usually considered as illiquid (for example, real estate and securities). Thanks to tokenization, investors can benefit from many more possibilities around the world because to the tokenization assets can be split into tokens and sold through digital markets.

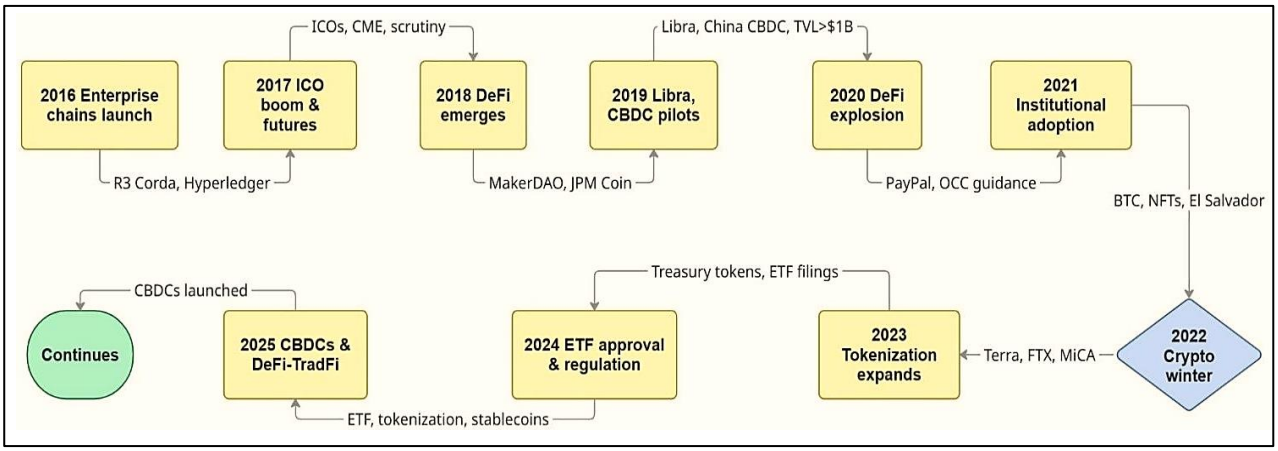


Figure – 3: Evolution Timeline of Blockchain and Digital Asset Innovation in the Financial Sector (2016–2025)

17.3.6. Insurance

The insurance industry is being revolutionized by blockchain technology, which enhances transparency, reduces the risk of fraud, and simplifies the process of carrying out complicated transactions in insurance. One of the most essential aspects of smart contracts is automation of the claims processing since smart contracts allow for automation of the process of claims verification, evaluation, and payment. Smart contracts are created based on reliable data connections and do not require any intermediaries who remove unnecessary stages of the process. This helps insurance companies save money, enhance customer experience, and provide necessary trust level due to the transparent and protected information. One more important use of the blockchain system is in parametric insurance, which functions on an automatic basis as soon as some parameters, like weather changes or flight cancellations, are attained. Thanks to the integration of AI and IoT sensors, the system is able to analyze data in real-time and start automatic operations with no need to go through the bureaucratic red tape involved. In agriculture, the implementation of blockchain technology in the realm of crop insurance and other financial services gives rise to an opportunity to achieve more precise estimation of losses, secure records of transactions, and more efficient collateral management.

17.3.7. Central Bank Digital Currencies (CBDCs)

Central Bank Digital Currencies (CBDCs) is one of the biggest institutional applications of blockchain technology and digital currencies. Simply they can be interpreted as secure digital currency created by the central bank. While CBDCs can basically be described as an efficient way of making transactions, allowing more access to the financial markets, and enabling

functionalities of programmable finance, CBDCs still ensure the same parameters in terms of stability and reliability as the traditional fiat currency. Therefore, the central banks now consider various CBDC designs, including retail and wholesale scenarios, thus implementing a modernized financial architecture. The wholesale basis of CBDCs could help in significant advancements in interbank settlements, asset management, and overall market efficiency when tokenized. The implementation of a CBDC despite being simple, can face problems like cybersecurity, privacy, stability and usability. The success of a CBDC rest on the balance between technological experimentation and high-level security, concrete regulation, and thereby ensuring access to all consumers.

17.3.8. Integrated FinTech Ecosystems

The influence of blockchain technology on the financial industry is a great challenge. When blockchain is merged with other technologies such as AI and electronic payment systems, it creates a unified ecosystem that has a significant role. With the integration of these technologies, one can access the financial services easily and it also increase the level of transparency. Secured nature of blockchain helps people to keep track of their transaction while keeping it safe. It also helps in fraud detection and automate final decision-making process on loans with help of AI. Blockchain in FinTech helps to reduce the gap between rural and urban people. It also allows payments to be quicker, lending to become more precise, and digital services to be offered to the underrepresented groups. Blockchain being the "immutable" data backbone combined with the power of AI provides real-time transactions tracking, better risk control, and enhanced compliance management. All of this together leads to the conclusion that the financial industry will become different in terms of inclusion and innovation in the future.

17.3.9. Case Study

ICICI Bank — Blockchain for Trade Finance and Cross-Border Payments

The implementation of blockchain technology at ICICI bank is meant to modernize trade finance practices and speed up cross-border payment transactions, eliminating unnecessary complexities that arise from documentation requirements, manual checks, and obscure settlement processes in relation to small and medium businesses whose operations become much more expensive as a result.

Problem: Traditional trade finance processes require a huge amount of paperwork and involve various types of intermediaries. Consequently, there are unacceptable processing times, high fraud risks, and restrictions on process transparency. Concerning cross-border payments, one has to consider long settlement periods and all the problems associated with correspondent banks.

Resolution: The bank successfully used blockchain based platform for trade to digitise documents used in trade transactions and to improve the speed of cross border payments.

Implementation timeline

- ***2017-2018:*** Exploration and testing with international countries to simplify the existing trade procedure activities.

- ***2019:*** Pilot results displayed improvement in documents processing times, accompanied by the faster remittances' testing.
- ***2020:*** Integration with core trade finance systems and development of partnerships allowing to implement blockchain solutions abroad.
- ***2021-present:*** Extending into the supply chain finance sector and constant optimization through experimenting with AI and IoT technologies.

Key outcomes

- The time taken for processing dropped significantly.
- The use of automated transactions will increase transparency and security.
- The operational cost has gone down considerably.
- Clients are satisfied with the service (due to greater accessibility of SMEs to trade financing).

Challenges and lessons: In terms of regulations, it is necessary to have legalization in place, thus it is good to get in contact with your regulators on day one. When merging blockchains with your legacy system, you will need custom-made adapters and phased implementation to make the process work fine. Moreover, you will also have to think about scaling up and onboarding the target partners from small businesses that may not be able to accept some changes all at once.

Relevance: In this case, ICICI's experience is a good example of how the current banks can employ blockchain technology to improve their activities in trade finance and make their operations more transparent ushering SMEs in the world of credits at the same time complying with the regulatory requirements.

17.4 Applications of Blockchain Across the Internet of Things (IoT)

In today's world, there is a huge advance in connectivity. Various devices like factories, home appliances, and health trackers produce and send raw data. The devices that are connected together form the Internet of Things (IoT). With billions of devices being connected worldwide, the central question is: How do we ensure the security and safety of access-control information? Traditionally, the Internet of Things (IoT) network has depended on central cloud servers to collect and process the totality of the incoming data. Although this method promotes the growth of the network, it carries a significant risk of all-over failure because of the functionality being affected by outages of a central server. Blockchain technology, which substitutes weak and unreliable central servers with safe and distributed digital system, turns out to be the right solution. Originally developed for cryptocurrencies, blockchain now enables machines to communicate with one another using technology of IoT. Through the combination of IoT's sensing capabilities and incorruptible ledgers of blockchain, independent, reliable systems may be built. This thesis pays attention to how the combination of these technologies helps in particular spheres.

17.4.1 The Structure of the Internet of Things (IoT)

Internet of Things acts as a digital nervous system, allowing objects to perceive the environment and communicate with each other through the transfer of information. Layers of IoT network include the following:

- **Physical Layer:** Layer of information gathering containing sensors ("eyes and ears" which sense the temperature, movements, humidity) and actuators ("muscles" which control the valve actuations).
- **Data Link Layer:** Layer which manages the transmission of information and error correction between physical devices and network nodes.
- **Network Layer:** Nervous system, transferring information through the technologies of Wi-Fi, Bluetooth, 5G, or LoRaWAN and ensuring data security when transmitting information from the physical device to the processor.
- **Transport Layer:** Provides reliable and permanent information transmission by handling the process of acknowledgments and sequencing of packets.
- **Session Layer:** Management of sessions and connection life-time.
- **Presentation Layer:** Layer which is also called the middleware layer where data is transformed, cleaned and analyzed. Nowadays, edge computing and Artificial Intelligence (AI) are frequently applied in this layer to analyze data locally and save time.
- **Application Layer:** User interface such as mobile applications and dashboards through which people would be able to manage their data and smart devices.

The major vulnerability of the classical architecture lies in the upper layers where the data is gathered and stored in the single centralized database.

17.4.2 Blockchain Technology Components

If the Internet of Things (IoT) can be compared to the nervous system of a living creature, blockchain may be considered its immutable memory. The term 'blockchain' refers to a digital distributed ledger. Information is not stored on a single central server but instead is kept in multiple copies of the ledger on thousands of separate computers called nodes. Imagine that there is a situation where all notebooks in the classroom are identical and an announcement is needed in order to add some new data. Each member of the classroom checks if the new data meets a set of certain rules. If everything is fine, the new data is added to everyone's notebook and no one is able to change or delete something without being detected by the rest of the class. In the case of blockchain technology, such data is represented by data logs from IoT devices. These data logs are placed in cryptographic "blocks" and connected in a sequence way. Any change in one block means that all following blocks have to be changed as well.

Smart Contract: Putting Automation in Trust

Smart contracts are programs designed to function automatically once certain requirements have been fulfilled. Just like the vending machine that works mechanically to give out drinks upon pressing buttons and inserting money without any human involvement, smart contracts also work in a similar way but in digital format. They perform certain actions depending on the stipulated requirements. Some of the activities performed by smart contracts include payments and alarms from sensor data.

Visualizing the Convergence

To comprehend how these technologies interact in real-time, the following diagram illustrates the flow of data in a modern Blockchain-IoT ecosystem.

17.4.3. Secure Storage and Data Management

Since there is no centralized place where data are stored, blockchain significantly increases the security level of data storage in the Internet of Things. However, storing large amounts of data like HD security video is not an effective way because of inefficiency both in terms of performance and costs. That is why, in a modern Blockchain-IoT system, a combination approach is used. All the data are stored offline in local databases; at the same time, the corresponding hashes are stored in a blockchain and act as fingerprints of this data. Any change of offline data will create a fingerprint different from on-chain fingerprints, which will trigger an alarm.

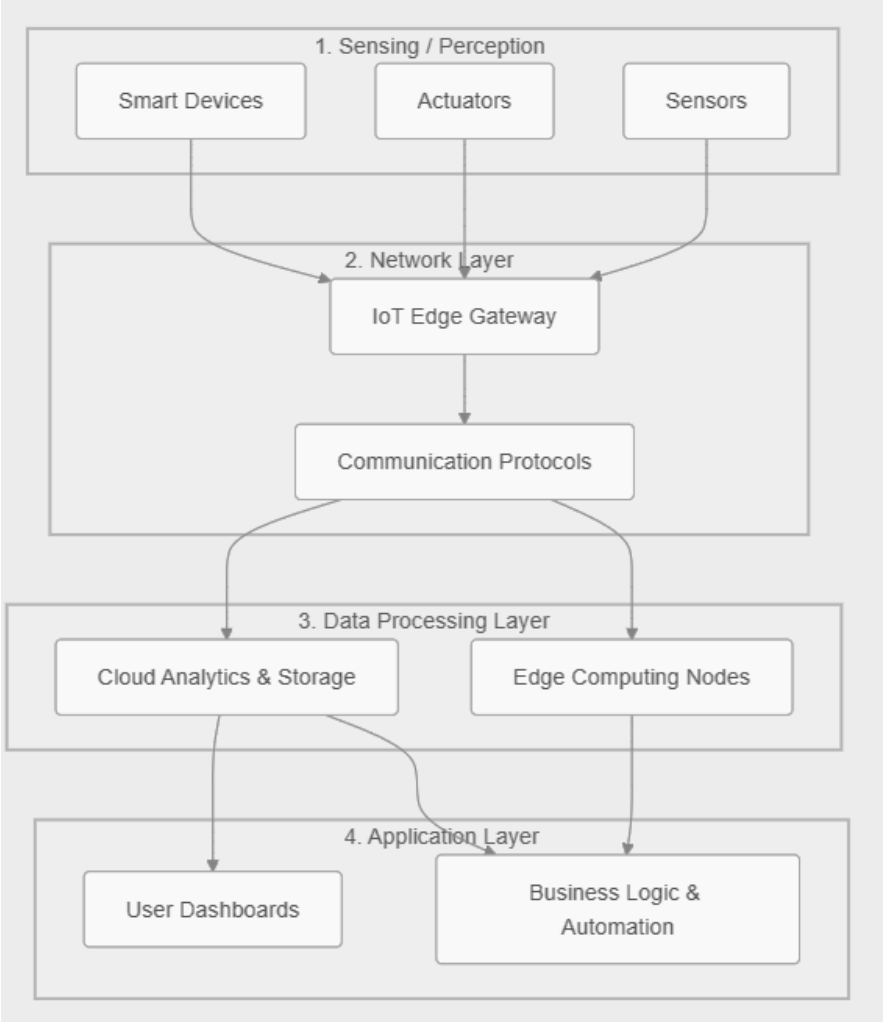


Figure – 4: Architecture of a converged IoT and Blockchain system

17.3.4. Supply Chain and Smart Agriculture

With the blockchain technology, there is easy tracing of goods from their point of origin up to their destination, solving many challenges involved in international trade by creating a common

ledger that can be viewed by producers, customs, and retailers. For example, when there is an electricity breakdown in the refrigeration system while vaccines are being transported, the IoT temperature sensor picks up this challenge and records it on the blockchain ledger. The smart contract then penalizes the transportation firm.

Table-1: A comparative analysis of traditional versus blockchain-enabled supply chain architectures.

Feature	Traditional Supply Chain	Blockchain-IoT Supply Chain
Data Storage	Databases controlled by individual firms.	Flexible ledger available to everyone who has authorization.
Verification	Traditional auditing practices with paper evidence.	Cryptographic hashes generated automatically based on IoT data.
Disputes	Prolonged debates with regard to conflicting documentation.	Smart contracts automatically imposing penalties for violations.
Security	Prone to fraud through retroactively modified logs.	Immutable since IoT data can never be altered.

17.4.5 Smart Agriculture 5.0

Crop insurance in traditional farming takes a lot of time and is prone to manipulation and frauds. In Smart Agriculture, farmers use IoT soil sensors and drones to measure the condition of crops. By connecting these IoT sensors to a blockchain, crop insurance becomes completely automated. When the trusted weather sensors detect a significant drought in a certain district, a smart contract automatically transfers digital money to farmers as relief funds.

Intelligent Autonomous Transportation: With the implementation of self-driving vehicles within cities, data security will be essential for safety reasons. These cars exchange their speed and information regarding roads with traffic signals and other cars. Thus, an intrusion into the central traffic server would allow hackers to alter information and cause accidents. Blockchain provides a way to protect this communication by validating traffic data in real time in a distributed ledger. Vehicles will be able to verify data sources before using this data. Furthermore, blockchain facilitates the deployment of self-executable smart parking and toll systems. The smart contract is triggered when the car enters a toll area, and the car's virtual purse gets in touch with road side sensors, and the service fee gets paid finally.

Security, Privacy, and Zero-Knowledge Proofs - The low computational abilities of cheap IoT devices like smart lights do not allow for effective antivirus security, making these devices vulnerable to misuse by attackers for launching large-scale attacks. This problem is solved using blockchain technology because the use of cryptographic keys instead of passwords guarantees that only the rightful owner can control the device. However, data aggregation brings about privacy issues. For instance, the smart electricity meter can reveal much information about the person's electricity consumption habits. In order to solve this problem, the Zero-Knowledge Proof (ZKP) approach could be used. The Waldo analogy is a perfect example of this method: a person is able to prove to his friend that Waldo is there without showing his exact location by

covering his face with a sheet of paper with a hole in it. The same principle applies to a smart house that can use ZKPs to prove to an insurance company that a security alarm was turned on during a burglary.

Healthcare and the Internet of Medical Things (IoMT) - Internet of Medical Things (IoMT) is made up of health-monitoring wearables, for example, continuous glucose monitors and smart pacemakers. The information obtained needs to be kept highly confidential while still easily accessible in times of emergencies. Smart contracts on the blockchain will allow linking wearables and sensors with EHRs. If a person experiences an arrhythmia detected by his/her heart monitor, a smart contract will decode his/her medical history and send it to the nearby emergency room in minutes. This will provide doctors with reliable information at their fingertips in real-time while maintaining the patient's privacy according to strict rules of healthcare industry and minimizing any potentially life-threatening bureaucratic delays.

17.5 APPLICATIONS OF BLOCKCHAIN ACROSS CYBERSECURITY

The blockchain is changing the dynamics of cybersecurity through providing decentralized, tamper-proof, and secure mechanisms that improve the authenticity, integrity, and threat detection capabilities. Blockchain technology was initially developed for cryptocurrencies; however, its essential attributes such as cryptographic security, immutability, and distributed consensus have been incorporated into solutions that would solve cybersecurity problems such as data breaches, identity theft, and cyber frauds. The use of blockchain with technologies like artificial intelligence and Internet of Things (IoT) brings about innovative possibilities for developing cybersecurity solutions. Utilization of the blockchain in the context of cybersecurity increases the reliability of the system and allows creation of trustless environment which makes cybersecurity an integral part of the system architecture.

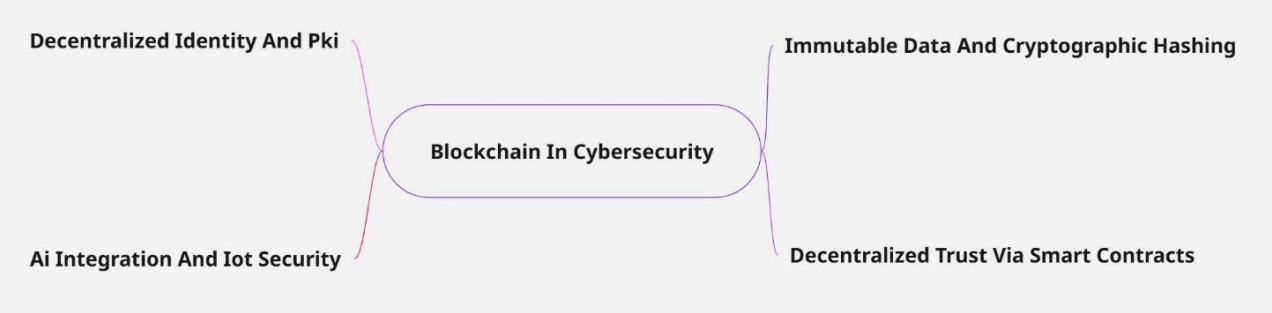


Figure – 5: The simplified overview - Identity and Access Management (IAM)

Self-Sovereign Identity (SSI) - Self-sovereign identity (SSI) leverages the power of blockchain technology to enable individuals to exercise greater control over their personal information by providing them with verified credentials that they can choose to share selectively. SSI minimizes reliance on centralized identification services, enhancing privacy and reducing the risk of data breaches. The verification of verifiable credentials stored on blockchain enables secure and reliable authentication mechanisms when interacting with services online.

Decentralized Authentication - The blockchain Authentication Systems will allow for Decentralized identity Verification where All identities Credentials are kept on record and Secure

in Distributed Ledger that cannot be tampered with or compromise since they're immune from any attacks because there are many Copies of it.

17.5.2 Data Integrity and Secure Audit Trails

Immutable Log Management

The unchangeable feature of blockchain ensures the creation of an unchangeable record of all incidents related to cybersecurity, system log, and transactions which is a great help in conducting full-scale forensic research due to the ensuing possibility of conducting audits and controlling the process of system monitoring in real-time as long as integrity is maintained.

Data Transmission *Security* The blockchain operates on the concept of the distributed system, permitting the intercommunication of various parties in a more convenient manner. To put this differently, in simple terms, the blockchain is a method of verification of information interchanged in a manner that can hardly be compromised.

17.5.3.: Threat Detection and Incident Response

- ***Distributed Cyber Threat Intelligence-***Distributed network using the power of the blockchain can be used to develop a platform for distributing cyber threat intelligence to different organizations.
- ***Automation via Smart Contracts -*** The use of smart contracts is necessary to perform an automated response, so the hacked resources would be blocked, user permissions would be revoked, and warnings would be provided timely.

17.5.4.: Secure IoT Ecosystems

IoT Device Authentication

- The use of blockchain technology enables the creation of a decentralized framework for the authentication and authorization process of IoT devices ensuring that these devices do not get mixed up with fake ones by virtue of high-quality cryptography. In addition to that, the technology ensures the storage of secure identities of devices which is crucial in industrial control systems.
- ***Data Integrity in IoT Networks -*** It is ensured by blockchain that messages sent to IoT system sensors as well as sent back messages remain the same from the time of transmission to the moment of storage. Compliance with data security regulations through blockchain-based cybersecurity
- ***Regulatory compliance -*** As transactions in the blockchain are immutable, compliance with different regulations for data security, such as GDPR or HIPAA, is simplified. Smart contracts make it possible to implement policies, which means that audit papers are generated automatically.
- ***Privacy-Preserving Data Sharing*** - Various crypto tools, like zero-knowledge proofs, can be used by companies as well on top of blockchain to ensure compliance with regulations while sharing information without exposing any confidential data as well.

17.5.5.: Prevention of Fraud and Insider Threats

- ***Transparent Transactions -*** The technology of blockchain features distributed ledger which minimizes insider threats and fraud due to the transparency, hence operation of transactions is visible to authorized users.
- ***Tokenized Privileges and Access*** - Tokenization of permissions and access rights has the ability to control access of users and avoid any abuse.

17.5.6.: Supply Chain Security

- ***Transparent Origin Verification -*** The ability of blockchain technologies to track the production of goods ensures safety in the supply chain and keeps it away from counterfeiting.
- ***Secure Contractual Terms and Payments -*** Automation of contracts with the help of smart contracts guarantees that all payments are done according to cybersecurity requirements.

17.5.7.: Smart Contract Vulnerabilities and Security Challenges

- In spite of the benefits provided by smart contracts through automation and trustless transactions, smart contracts are still vulnerable to certain security issues. Possible problems arising from the vulnerabilities of the smart contracts include:
- **Re-entrance attacks** which happen due to recursion of functions resulting in any manipulations with the money or smart contract state.
- **Arithmetic mistakes** such as integer overflow and underflow that might cause the bugs in functioning of smart contracts.
- **Bad access control** – lack of proper permissions allowing to perform certain actions for unauthorized parties.
- **Denial of service attack** aimed at overloading the target or exploiting its vulnerability to become unavailable.
- Lack of **randomness** and **oracle problems** influencing the output of smart contracts.
- **Bugs and vulnerabilities** that can be missed during the smart contracts' code auditing process.

To prevent the possible issues, smart contracts should be audited and formally verified.

17.5.8.: OWASP Top 10 Vulnerabilities in Blockchain and Smart Contract Ecosystem

The OWASP (Open Web Application Security Project) Top 10 highlights the most critical security risks in web and blockchain applications, including smart contracts. Key vulnerabilities relevant to blockchain include:

- **SC01 – Access Control Vulnerabilities:** Weak access restrictions let in unauthorized users, they can end up running privileged functions like minting burning upgrading contracts, or even withdrawing funds.
- **SC02 – Business Logic Vulnerabilities:** Bugs in the contract's rules or its economic design, make it possible for adversaries to tweak lending staking governance, and reward processes while still staying inside what the code technically permits.

- **SC03 – Price Oracle Manipulation:** Attackers mess with outside price feeds or on chain liquidity pools, so asset prices get reported wrong. This can cause messed up collateral valuation, unfair liquidations, or too much borrowing.
- **SC04 – Flash Loan-Facilitated Attacks:** By using flash loans without collateral, an attacker can grab big amounts of cryptocurrency inside a single transaction. Then they leverage issues like oracle manipulation, or business logic flaws, to walk away with profit.
- **SC05 – Lack of Input Validation:** If user inputs, addresses, transaction parameters, or external data aren't checked properly, malicious values can slip through and shift smart contract behaviour away from what was intended.
- **SC06 – Unchecked External Calls:** Smart contracts that initiate communication with other contracts without verifying that outside activation was actually successful may have peculiar and unexpected stops or unusual conditions and also may suffer from callback vulnerabilities.
- **SC07 – Arithmetic Errors:** When math calculation fails to produce results or the mathematical operation results in rounding or when decimals of tokens are treated incorrectly, the balances can be miscalculated or the distributions of the profits can become unequal.
- **SC08 – Re-entrancy Attacks:** The malicious smart contract can keep calling the faulty function multiple times before its first run finishes, leading to repeated withdrawal of fund or altering the contract's state.
- **SC09 – Integer Overflows and Underflows:** When calculation fails to be handled properly, the numbers can go above or below the limits and lead to wrong balances and even wrong creation of funds.
- **SC10 – Proxy and Upgradeability Vulnerabilities:** Inappropriate proxy configuration, insecure contract initiation procedures and lack of regulation regarding rights acquisition may lead to criminals obtaining control over the firm.

17.6 FINAL THOUGHTS

The blockchain system is clearly positioned to set new records for the most advanced and outstanding system. It opens new horizons to automation and technological advances in various areas. The first reason for the use of blockchain technology in different industries is security, as its main goal is to enhance security. In addition, blockchain technology serves as an effective tool for the approximation of IoT Technologies in terms of developing intelligent systems and creating devices equipped with advanced features. Similarly, the technology makes it possible to shorten the processes of carrying out transactions and promoting activities related to the digital economy. Blockchain usage in the industry comes helpful in terms of managing time, providing transparency in transactions, and increasing effectiveness in production. Thus, the implementation of blockchain technology can have a significant impact on many spheres.



Figure- 6.: OWASP Top 10 Smart Contract Vulnerabilities

At the same time, there are certain problems connected to it, namely rules and regulations that are not developed yet. It is particularly important to address legal issues connected to this system due to the fact that false or bad use of the technology can lead to its collapse.

CHAPTER 18

FUTURE DIRECTIONS: DECENTRALIZED AGI, QUANTUM-RESISTANT SYSTEMS AND ETHICAL HORIZONS

Contributed by: Sourav Saha, Sangita Bose, Sushovan Paul

18.1 INTRODUCTION

Rapidly converging technologies such as blockchain, AI, distributed computing, and enhanced cryptography are transforming the future of intelligent secure ecosystems. Although current AI applications are very effective in perceptual, reasoning, linguistic, and decision-making tasks, their implementation has been mostly centralizing, resulting in issues concerning transparency, trust, scalability, data ownership, and governance. At the same time, blockchain has proven itself to be an emerging decentralized computing paradigm capable of delivering immutable data integrity, distributed consensus, identity management, and programmable trust via smart contracts. As a result, the combination of the aforementioned technologies has resulted in a completely new class of decentralized intelligent ecosystems. In this chapter, we will look into future trends in developing blockchain-based intelligent systems, including but not limited to decentralize Artificial General Intelligence (AGI), blockchain infrastructures that are robust against quantum attacks, and ethical considerations for autonomous decentralized societies. First, we will review the development of intelligent computing from centralized system designs to decentralized AI ecosystems, and see why blockchain is important for the emergence of decentralized intelligence. Next, the principles of decentralized AGI and how it is different from its centralized counterparts are explained, along with possible benefits of such technologies in terms of democratizing intelligence, and associated problems with monopolies, governance, and computational fairness. We will also explore emerging paradigms of cooperative multi-agent systems, decentralized learning algorithms, incentive schemes based on tokens, and decentralized AI marketplaces via blockchains.

The field of Artificial Intelligence has changed a lot. It used to be an area of research but now it is one of the main technologies that affects our society. In the ten years, improvements in deep learning foundation models and big computers have made smart machines much more possible. Artificial Intelligence is now used in areas like science, medicine, business, industry, transport, education, and cybersecurity and software development. However most current Artificial Intelligence systems are still. Managed in a centralized way. Language models, multimodal reasoning engines and machine learning frameworks are usually created, trained

and used with cloud computing resources owned by a few companies. This has helped technology advance. It has also caused problems with transparency, accountability, privacy, accessibility and unfair social differences. Artificial Intelligence systems need to be more open and fair. The use of Artificial Intelligence must be transparent and accountable. Artificial Intelligence should be accessible, to everyone. The development of Artificial Intelligence must consider disparity. We need to make sure Artificial Intelligence benefits everyone. There is another approach that could be used to solve these problems and is based on the concept of blockchain. While traditional centralized technologies do not allow for decentralized coordination, blockchain does. The key differences between blockchain and other technologies are cryptographic validation, consensus, an immutable ledger, and smart contracts. Unlike traditional approaches where one needs to find trusted intermediaries to maintain trust, blockchain builds trust using mathematical proof and consensus between participating nodes. It is against this background that the convergence of blockchain and AI has emerged as an essential area of academic research. Blockchain offers the elements of security, transparency, provenance, and decentralized governance, while AI provides reasoning, learning, adaptability, and autonomous decision making. The combination of these two technologies offers a basis for intelligent decentralized systems to enable collaboration in the creation of knowledge and learning. Among the numerous goals that have arisen due to the emergence of such synergy is the development of Decentralized Artificial General Intelligence. While narrow Artificial Intelligence systems are built specifically for the purposes of accomplishing particular tasks, AGI aims to create computational systems with the capabilities to reason, adapt, learn from previous experience, and solve various problems. Although most of the current work in AGI concentrates on creating centralized computational systems, the decentralized approach presupposes the development of intelligence in agent collaboration networks. The future of intelligent ecosystems will also be shaped by developments in quantum computing. While quantum technologies will bring about revolutionary possibilities for computing power, they also pose a threat to many cryptographic protocols protecting blockchain infrastructure. Therefore, future decentralized AI systems need to adopt quantum-resistant cryptography that can guarantee security from future computational threats. Apart from technical considerations, the implementation of autonomous intelligences within decentralized governance poses many ethical concerns. Intelligences that can think and act independently, and participate in economic activities will have an impact on social organizations, public policy, financial markets, science, and digital governance. Being able to create such systems that are transparent, accountable, just, and humane is the key challenge of the coming decades. This chapter investigates the relationship between these developments through discussions about AGI architecture that is decentralized, blockchain-based intelligent ecosystems, quantum-proof infrastructure, and ethical frontiers for autonomous decentralized societies. The objective of this investigation is to present a futuristic perspective of the technologies that can possibly constitute the future of intelligent computing.

18.2 EVOLUTION OF BLOCKCHAIN AND INTELLIGENT ECOSYSTEMS

The emergence of intelligent ecosystems is defined by an ongoing interrelation between advancements in computational intelligence and distributed information technologies. Thus, we will get a good insight into why the concept of blockchain is so significant for future

intelligent infrastructures. Earlier computer systems were predominantly stand-alone computational devices programmed to perform deterministic computations. The computation was performed by means of certain programmed tasks on a single device, which did not have any adaptation abilities and autonomous control. In case there were any smart features in such systems, these features were intentionally built into the software code by people. The emergence of computer networks changed the situation radically due to the introduction of the ability to distribute computations. Companies used the capabilities of networked computers to implement collaboration, distributed databases, and distributed computations, but the centralized way of management was still dominating. The development of cloud computing was a big deal. Cloud computing made it possible to get computing power when you needed it so you could train complex machine learning models and use artificial intelligence solutions all around the world. Cloud computing had some features like being able to manage resources in a flexible way using virtualization technology and storing data in lots of different places, which made it much faster to do artificial intelligence research and put it to use in real world situations. Cloud computing helped speed things up for machine learning models and artificial intelligence solutions. Yet, cloud-centric solutions were contributing to further centralization of control over computing resources and data. Companies working on advanced AI solutions created immense databases of user data, computing resources, and proprietary models. All these technological capabilities posed issues of privacy, transparency, digital sovereignty, vendor lock-in, and access to AI solutions. Blockchain technology has been developed as a reaction to numerous structural problems. Firstly, it became clear that blockchain could be used not only as a foundational technology behind cryptocurrencies but also as a decentralized platform for conducting secure digital transactions. The core innovation that blockchain has introduced is in the distributed ledger system architecture. Each participant holds an exact copy of the distributed ledger. At the same time, the consensus algorithm ensures that there is a common understanding of valid transactions. Cryptography, digital signatures, and consensus algorithms guarantee data integrity, transparency, and tamper-proofness. These attributes are not confined to just financial use cases. The blockchain technology can support identity management, supply chain tracking, asset ownership, voting, health care records management, intellectual property protection, and scientific research collaboration. On the other hand, AI has also experienced a significant revolution due to advances in machine learning and deep neural networks. In particular, as opposed to earlier times when AI relied purely on symbolic reasoning, the modern approach involves learning from large volumes of data directly. Foundation models and Large Language Models have really taken intelligence to the next level. These models are great at understanding language, generating code, reasoning, summarizing, translating and having conversations. What is really cool about them is that they can do a lot of things.

The idea of combining blockchain and artificial intelligence was bound to happen because these two technologies have been growing side by side. Blockchain is good at keeping things safe and decentralized, and artificial intelligence is good at understanding information and making decisions on its own. Blockchain and artificial intelligence together can do a lot of things that blockchain and artificial intelligence can do separately. Blockchain and AI convergence resulted in intelligent ecosystems that have the following features.

1. The intelligence is spread out not controlled from one place. Intelligent agents can be made to work in networks where no single server does all the computing. Trust becomes algorithmically verified instead of institutionalized. There is no need to refer to centralized authorities because blockchain consensus allows verification of everything transparently.

2. Governance can be made programmable, with contracts. They help organizational policies work on their own. This is done through decentralization, which also enables incentives and collaborations to operate. Governance and smart contracts make this possible. Organizational policies and decentralization are key here. The use of contracts allows Governance to function independently.
3. Knowledge sharing is made reliable by blockchain records. These records keep track of where datasets, AI models, workflows and decisions come from. This information is stored forever. As a result, future smart systems will likely stop using AI. Instead, they will use a network of computers. In this network, independent agents, groups, and people work together. They use shared resources to get things done. The system is based on blockchain entries. This way, knowledge sharing is transparent and trustworthy. It helps build a community of agents. They work together with organizations and individuals. The goal is to create a way of computing. It is called a society.

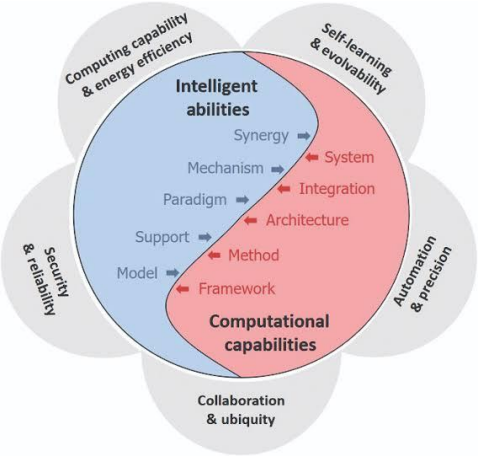


Figure 18.1: Evolution of Intelligent Computing Ecosystems

18.3 ARTIFICIAL GENERAL INTELLIGENCE: CURRENT STATE AND FUTURE VISION

Artificial General Intelligence is a big deal and a huge challenge in computer science. Usually, people make Artificial Intelligence systems that can only do one thing like recognize pictures or translate languages. But Artificial General Intelligence wants to make machines that're smart in a general way so they can do lots of things without being told exactly what to do. The difference between Artificial Intelligence and Artificial General Intelligence is important. Regular Artificial Intelligence is good at doing tasks like recognizing pictures, translating languages, helping doctors diagnose illnesses or suggesting things we might like. These systems are really good at what they do, sometimes even better than humans who're experts in those areas. Artificial General Intelligence systems have a hard time taking what they know and using it in a completely different area or switching to a new task on their own. Artificial General Intelligence is still trying to figure out how to make machines that can do this. However, general intelligence includes such cognitive skills as reasoning, planning, abstract thinking, causal reasoning,

creative thinking, learning, memory, communication, and flexible problem solving. This can be seen in human intelligence as people can use knowledge gained to solve new problems and learn all the time. Existing artificial intelligence algorithms possess certain key traits of generalized intelligence. First, large language models show linguistic ability, contextual thinking, coding, and knowledge synthesis abilities across various fields. Second, reinforcement learning algorithms are starting to develop planning skills within complicated environments. Third, multimodal algorithms learn to combine visual, text, and audio information in unified representations for richer reasoning processes.

The current AI lacks some of the essential abilities of generalized intelligence. First, most of the AI systems do not have long-term memory. Second, they cannot reason causally and autonomously set goals. Third, there is no lifelong learning and deep situational awareness. The knowledge of existing AI is mostly statistical in nature. These limitations inspire further research in the field of building AGI architectures. Traditionally, AGI research has focused on centralized computational infrastructures consisting of large training data sets, computational clusters, and optimization processes. However, while these methods have been highly successful, they also centralize intellectual capital, computational power, and governance to a limited number of organizations. The centralization of AGI research poses several social problems. Control over advanced AGI technology could be controlled by a select few organizations that limit access, transparency, and the influence of technology. Centralized AGI infrastructure could also be vulnerable to threats of cybersecurity, system failure, regulatory capture, and monopoly. These issues have fueled increasing interest in decentralized AGI systems. Instead of building a centralized superintelligence, decentralized AGI involves building intelligence through the collaboration of multiple computational agents within blockchain-based environments. Within a decentralized framework, each agent would be able to contribute certain competencies including reasoning, planning, perception, knowledge lookup, optimization, negotiation, science discovery, or resource management. The use of blockchain is required for a coordination infrastructure for communication, governance, identity, and knowledge exchange between the participating agents. These systems are closer to the nature of biological and social intelligence, where complicated collective actions are performed by the interaction of many specialized individual units without central control. Human societies, science, markets, and ecosystems provide examples of distributed intelligence created as a result of the collaboration of different independent agents.

In light of these observations, we can assume that future AGI systems will be developed based on the model of decentralized cooperation between agents instead of traditional centralized solutions. This approach is promising in terms of scalability, robustness, transparency, adaptability, and democratization of decision-making processes. On the hand decentralized Artificial General Intelligence or AGI brings up some challenges. These challenges include consensus algorithms, coordination efficiency, and alignment of interests, interoperability, security and ethics of decision-making with AGI. To solve these problems, we need research in these areas. These areas are intelligence, distributed computing, blockchain technologies, economics, cybersecurity, and computational social sciences. So moving towards AGI is not just about technology evolving. It is also a step in organizing and governing intelligence in digital societies with AGI.

18.4 DECENTRALIZED AGI ARCHITECTURES

Artificial General Intelligence is a computer system that can think about everything. People thought that if they made the computer programs smarter by training them on a lot of data then the computer would become intelligent. They did make some progress with this method. Were able to create some really good models and language programs. However Artificial General Intelligence has some problems. Artificial General Intelligence is not very good at handling a lot of work. Artificial General Intelligence is also not very clear about how it makes decisions. Not everyone can use Artificial General Intelligence because it needs a lot of computer power. Artificial General Intelligence has some issues with control and, with staying safe. Artificial General Intelligence is the thing that people are trying to make. The new paradigm of intelligence implies that intelligence is not an attribute of a single computational system, but the emergent result of interactions of autonomous agents within decentralized environments. Decentralized AGI implies transition from centralized cognitive architectures to distributed intelligence where specialized computational agents perform different tasks and interact for solving problems at different levels of autonomy. Within the framework of decentralized AGI, specialized agents could have their own fields of competence. For example, one agent is able to reason logically, another to find scientific facts, another to make plans and optimize results, while other agents are responsible for perceiving the world, communicating with each other, securing the system from cyber threats, performing financial calculations, or controlling the process autonomously. Blockchain technology can provide the required trust foundation to orchestrate distributed intelligence. Each of the agents involved will have a crypto-secure identity that would ensure authentication without the involvement of any centralized entities. The smart contract defines the interaction protocols to govern cooperation, resource allocation, information transfer, and incentivization. This architecture has a number of significant benefits.

1. Scalability is greatly increased due to the ability to distribute computational loads among multiple independent nodes rather than having them processed within centralized data centers. With new computational power becoming available, new agents could enter the ecosystem without any changes to the architecture itself.
2. Resilience is provided by the decentralized nature of the system. Classical centralized AI-based systems are highly vulnerable to hardware malfunctions, cyber-attacks, and disruptions of services in centralized infrastructure. Decentralized agent ecosystem solves the problem of single point of failure, thus enabling operations even if certain parts fail.
3. Modularity allows for continuous improvement. Agents may change individually thanks to specific optimization and training while still being able to interact with the rest of the ecosystem. It is possible to implement gradual innovations without retraining the entire AGI system.
4. Decentralization of governance allows for transparency and democratization of the process. Rather than having all decision-making powers within the organizations, the governance can be performed by communities via DAOs.

However, there are many engineering issues involved in decentralized AGI. The need to coordinate between thousands or millions of independent agents in order to ensure efficiency necessitates a communication system that is both low latency and consistent. There is also an issue of managing resources when computational demands vary dynamically over the network. In addition, ensuring global consistency while retaining local independence is an unsolved problem in distributed AI.

Therefore, decentralized AGI can no longer be considered an architecture but a new philosophy about intelligence.

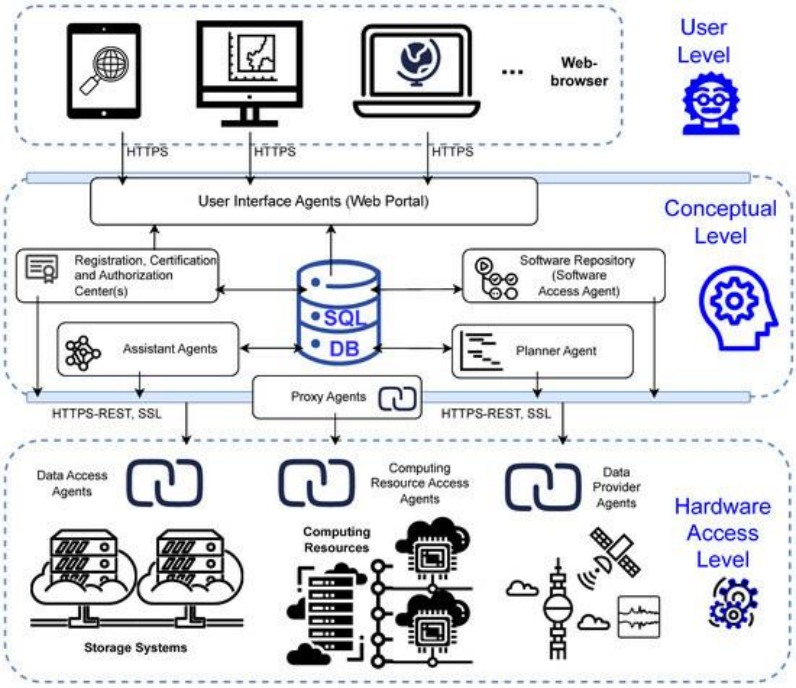


Figure 2: Conceptual Architecture of a Decentralized AGI Ecosystem

18.5 BLOCKCHAIN AS INFRASTRUCTURE FOR DISTRIBUTED INTELLIGENCE

Blockchain has advanced from being a mere technology platform for cryptocurrencies to becoming an infrastructure platform for reliable distributed computation in diverse ecosystems. Intelligent ecosystems in the future would require blockchain not just for making monetary transactions but also for identity management, decentralized governance, knowledge sharing, computational accountability, and collaborative intelligence. Decentralized trust is one of the greatest benefits of blockchain in intelligent ecosystems. Traditional artificial intelligence solutions often depend on central databases, authentication servers, and authoritative administrative control to enable computing. This results in vulnerability due to the presence of single point of failure, manipulation, censorship, and institutional control. Blockchain is really good at solving a lot of problems because it uses a group of people to agree on things of one person in charge. Every single transaction, every time someone does something on the computer or every time someone shares information gets stored in a safe database that nobody can change. This database uses codes to keep everything secure. So, people can share information with each other without knowing each other. One of the things about Blockchain is

that it helps us know where our information comes from. New artificial intelligence systems need a lot of data from places. We need to know if this data is real if somebody changed it who owns it and who can use it. Blockchain helps us keep track of all this information so we can see everything clearly. It also helps us keep track of how our artificial intelligence systems are trained what changes we make and how we use them. We can also use something called contracts to make things happen automatically. Normally, we need someone in charge to make sure everyone does what they promise. With smart contracts things happen on their own when certain things are done. So, people can make agreements with each other, share computer power, give rewards and check each other's work. Blockchain also has a way of rewarding people for working together. We can give people tokens for sharing computer power, data or special artificial intelligence models. This makes people want to work and share things with each other. Blockchain is really good at helping us do all these things. Blockchain makes it easy for people to trust each other and work together. We can use Blockchain to make our artificial intelligence systems better and more secure. Blockchain is a powerful tool that helps us build trust and work together. Another way in which blockchain can boost cybersecurity is by enhancing the capabilities of intelligent environments. Through distributed storage, encryption, decentralization of authentication, and unalterable audit trails, blockchain increases the level of security from interference and manipulation. Nevertheless, there is still a need for further development of blockchain technology in order to meet the demands of future intelligent environments. The current consensus algorithms used in blockchain often cause some delays and energy consumption and may hinder the coordination of AI in large-scale projects.

Table 1: Contributions of Blockchain to Future Intelligent Ecosystems

Blockchain Capability	**Contribution to AI Ecosystems**
Decentralized Identity	Secure authentication of autonomous agents
Distributed Consensus	Trustworthy coordination among agents
Immutable Ledger	Transparent audit trails and provenance
Smart Contracts	Automated collaboration and governance
Tokenization	Incentive mechanisms for decentralized learning
Cryptographic Security	Protection against unauthorized modification
Distributed Storage	Resilient knowledge management

The table illustrates how blockchain extends beyond financial applications to provide multiple foundational services supporting decentralized intelligent ecosystems.

18.6 AUTONOMOUS MULTI-AGENT ECOSYSTEMS

The future of artificial intelligence seems to be increasingly associated with multi-agent collaboration ecosystems rather than standalone intelligent systems. Multi-agent ecosystems represent a computational framework that allows numerous intelligent entities to collaborate, compete, negotiate and coordinate their actions in order to pursue certain goals. Every single

autonomous agent in such ecosystems has its own knowledge, skills, goals and decision-making algorithms. These agents interact with their environments constantly while exchanging data via secure communication infrastructure. Moreover, their joint activity is often characterized by emergent properties that are superior to the capabilities of their parts. Blockchain provides an excellent tool for coordinating the activity of multi-agent ecosystems since it guarantees secure communication channels, decentralized authentication, transparency of the governance system and programmability of interactions. Instead of using centralized coordination servers, agents communicate via blockchain-based infrastructure. Several forms of collaboration are expected to characterize future decentralized intelligent ecosystems.

1. Knowledge-sharing agents contribute specialized expertise while continuously updating distributed knowledge repositories.
2. Planning agents coordinate complex workflows involving numerous interconnected tasks.
3. Negotiation agents allocate computational resources through decentralized economic mechanisms.
4. Verification agents validate information quality, computational integrity, and regulatory compliance.
5. Security agents monitor network activities while detecting malicious behavior and responding to cyber threats.
6. Learning agents continuously improve local models while contributing generalized knowledge to collaborative learning frameworks.

This is similar to biology, where intelligence arises out of the collaboration of specialized cells, organs, and organisms, rather than centralization of processes. Similar principles are at work inside human organizations, science, and economies. Distributed intelligence helps in being adaptive. In cases of changing environment, specialized agents might change their behaviors independently while ensuring that there is no disruption to the working of the ecosystem. Some applications of decentralized multi-agent ecosystems include various fields. Science research could consist of autonomous agents working together to form hypotheses, experiment design, results analysis, and validating results. Ecosystems in healthcare might consist of coordination of diagnostic agents, treatment plan formation agents, patient monitoring system, pharmaceutical database and hospital resource management system. Smart city ecosystems might consist of transportation agents, environmental monitors, energy management systems, emergency response coordinator, and public services optimizer agents. Industry ecosystems might include robots for manufacturing, predictive maintenance agents, logistics coordinators, supply chain manager, and quality assurance systems all working in a coordinated manner. It can be seen from these examples that intelligent ecosystems of the future would look like a digital society made up of specialized autonomous computational entities.

18.7 QUANTUM COMPUTING AND BLOCKCHAIN SECURITY

While blockchain ensures good security properties based on modern-day computational assumptions, the rise of quantum computing presents a paradigm shift which could have

profound implications on cybersecurity. Quantum computing is an entirely different beast when compared to classical computing. Quantum computing takes advantage of quantum mechanics to compute certain functions faster than classical computers can achieve through classical algorithms. With ongoing research and development into fault-tolerant quantum computers, it seems likely that there will be quantum computers capable of challenging the security of many cryptographic systems in use today. Many public-key cryptosystems in use today, which are used in blockchains, like RSA and ECC, rely on certain mathematical problems that are believed to be infeasible to solve using classical computers. Shor's algorithm, which has been proposed as one of many quantum algorithms, has proven the ability to efficiently solve those problems. Such issues pose serious consequences for intelligent ecosystems powered by blockchains.

i. Identity fraud may allow adversaries to assume the guise of autonomous agents.
ii. Transaction validation can lead to a compromise of decentralized governance systems.
iii. Smart contracts can be subject to forgery with regards to their authentication process.
iv. Vulnerable signatures within historical blockchain transactions may be questioned with regards to their authenticity.
v. Large-scale decentralized AI infrastructure thus requires pre-emptive transition to quantum-proof security systems.

Fortunately, blockchain features inherent properties that would allow its cryptographic system to evolve incrementally. Currently, there is extensive research being done on blockchain architectures that would allow quantum resistance. These blockchain architectures include hybrid cryptosystems, which combine classical and post-quantum cryptography, adaptive consensus algorithms that would facilitate cryptographic updates, decentralized key management systems, and quantum-resilient identity systems. Thus, the combination of quantum computing and blockchain technology brings both problems and prospects. While quantum computers bring new challenges for cryptography, they also inspire innovations for creating secure, flexible, and resilient intelligent systems. This issue will be discussed further in the next section, which is devoted to post-quantum cryptographic systems.

18.8 QUANTUM-RESISTANT CRYPTOGRAPHY FOR INTELLIGENT ECOSYSTEMS

The rise of quantum computing has brought about one of the greatest shifts in the realm of cybersecurity. While the blockchain has proved itself to be highly resistant to classic computing attacks, the predicted powers of large quantum computers mean that new cryptographic approaches have to be devised which will ensure long-term security. In order for decentralized intelligent ecosystems, relying on secure communications, trusted identity management, and immutable records of transactions, to function safely, the use of quantum-resistant cryptography becomes more than just a necessary step – it is an absolute imperative. Post-quantum cryptography is defined as the set of algorithms that are designed to remain secure from attacks by both classical and quantum computers. Unlike conventional methods of public key cryptology, post-quantum approaches utilize mathematical problems that remain unsolvable in the face of quantum algorithms. Such problems include lattice-based approaches, code-based cryptography, multivariate polynomials, hash-based digital signatures, and isogeny-

based cryptographic approaches. While all these methods possess their own advantages and disadvantages, together they can become the cornerstone of future digital security. In blockchain-based intelligent ecosystems, quantum cryptography is expected to affect various aspects of the architecture. The cryptographic signatures used in the identities of autonomous agents should resist quantum attacks. Furthermore, smart contracts that govern decentralized cooperation should take place in a cryptographically secure environment, which ensures that the integrity of the system remains intact even as quantum computing becomes more advanced. In addition, consensus mechanisms might need to change to include quantum-proof authentications. One other area that needs to be considered when implementing post-quantum infrastructures is data privacy. Numerous blockchain systems store data that are required to stay valid for decades into the future, such as health records, legal papers, finances, scientific discoveries, and proof of ownership of digital assets. Even though quantum computers that would be able to break the current encryption might not be developed in the coming years, attackers might already be accumulating the data so that they can decrypt it when technology develops to allow it. There are some engineering problems involved in converting existing blockchain ecosystem into quantum resistant architectures. The new cryptography algorithms tend to use large key sizes, higher computational intensity, and increased communication costs than the conventional algorithms. Such elements can affect the transaction speed, data storage, network latencies, and power consumption. For that reason, some researches are done on optimized versions that provide better trade-offs between security and performance. One of the possible methods for migrating existing blockchains into quantum-resistant architecture is the development of hybrid cryptography architectures. It implies that there is no need to substitute existing cryptographic tools immediately but to introduce both classical and post-quantum algorithms in the intermediate stage of migration. However, there is another problem associated with the usage of post-quantum cryptography which is related to the creation of interoperable protocol standards for secure global intelligent ecosystems. The work of governments, academia, industries, and standardization organization would play significant role in solving this problem in the future decades. Quantum resistant cryptography must be regarded as more than a protective tool; it must become an essential part of the decentralized intelligence of the future. The creation of secure and autonomous societies needs cryptographic systems that can maintain trust even as computational power keeps improving.

18.9 ETHICAL HORIZONS OF DECENTRALIZED INTELLIGENCE

The development of technologies towards decentralized AGI raises ethical considerations that go beyond typical considerations in the field of artificial intelligence. Intelligent ecosystems working on the basis of decentralized architecture have the possibility of having an impact on economic systems, science, medicine, government, education, and management. AI researchers have a job to make sure that systems like these do what people think is right. When we talk about systems that're smart and work on their own but are also spread out across many different parts we have to think about ethics in a different way. We cannot just blame one company if something goes wrong. There are people and groups involved like the people who build the systems the companies that provide the infrastructure the systems that work on their

own the groups that make the rules and the people who use the systems. People need to be able to see what is going on in these systems. This is called transparency. The smart agents should keep records of how they make decisions so we can check that they are doing the thing. Blockchain is very helpful here because it keeps a record of everything that happens, and this record cannot be changed. We have to be careful because we also need to protect people private information. Sometimes these agents have access to information about people or companies and we need to make sure that this information is kept safe. Decentralized intelligence needs to have transparency and AI researchers need to make sure that decentralized intelligence is used in a way that's fair and right. Transparency is very important, for intelligence and it is something that AI researchers need to think about when they are building these systems. We need to think about fairness. When these smart systems start to control things like money, education, healthcare, jobs and government services, the biases in the systems can cause problems that will be very hard to find and fix. This is because these systems can work on their own without people overseeing them. So we need to make sure that these smart systems have ways to check themselves all the time.

We also need to think about privacy. Blockchain is good at keeping data safe. It does not work well with rules that say we need to be able to change or delete personal information. There are some tools that can help with this, like zero-knowledge proof, secure multiparty computations, homomorphic encryption and decentralized identity management. These tools can be very useful for ecosystems like blockchain. Intelligent ecosystems, like these need to be able to protect people's privacy. One further ethical concern is related to human autonomy. With intelligent agents taking on more complex decision-making tasks, retaining human agency becomes crucial. Decentralized autonomous systems need to enhance, not replace, human decision-making in areas where there is considerable moral, legal, or social impact. Designing such systems around human considerations is thus integral to proper decentralized intelligence. The prospect of decentralized AGI raises philosophical concerns about machine agency, moral accountability, and computational rights. While at present AI systems are just computational devices rather than conscious beings, future progress could undermine existing ideas about intelligence and agency. Specialists from computer science, philosophy, cognitive science, law, and ethics would thus be increasingly involved in the development of future frameworks of governance. Sustainability is one more ethical aspect. Decentralized blockchain and AI architectures use considerable amounts of computing power. Future decentralized intelligent systems have to incorporate energy-efficient consensus algorithms, sustainable hardware designs, and sustainable computation. Firstly, inclusivity should continue to be a core principle. Decentralized infrastructures that are intelligent should not be accessible by highly technologically savvy entities or economically advantaged groups alone. Democratic participation and equitable distribution of resources would play a very big part in deciding whether the future intelligent systems will foster inclusiveness in society. This is why ethics need to be considered during the designing and evolving of these decentralized AGIs.

18.10 GOVERNANCE MODELS FOR AUTONOMOUS DECENTRALIZED SOCIETIES

The system of governance is known to rely on central organizations that create rules, settle disputes, distribute resources and earn the public trust. Intelligent decentralized ecosystems are likely to go against many established norms concerning the system of governance in terms of distributing power among blockchain, autonomous systems, and collaborative groups. Programmable forms of governance based on smart contracts may become more common as time goes on in the decentralized world. Governance through such mechanisms allows executing organizational policies, voting processes, financial rules, and agreements automatically according to computational algorithms. This way of organizing the process allows for reducing the bureaucracy and increasing consistency. Decentralized Autonomous Organizations (DAOs) can be called one of the biggest breakthroughs in the sphere of governance. They allow communities to coordinate their decisions without any central control. Members of these communities create proposals, vote for organizational policies, resource distribution and ecosystem development. In decentralized AGI systems, governance is not confined only to humans but includes autonomous computing entities. Autonomous intelligent agents can manage resource sharing, verify knowledge inputs, collaborate on science, and improve infrastructures based on governance rules that are defined by decentralized agreements. Nevertheless, autonomous governance poses certain problems as well. For instance, there may be conflicts between optimizing algorithms and human values. In such case, governance mechanisms have to allow for human override. Hybrid systems, which combine computational efficiency with democratic control, can be one solution. Interoperability of regulations is another issue to take into account. Ecosystems of the future will consist of numerous jurisdictions with their own laws, regulations concerning privacy and cybersecurity. Dynamic governance structures that can adapt to changes in technology will be necessary. Static governance frameworks could very well become outdated as the decentralized AI ecosystem grows more complex. Evaluation of the policies, involvement of stakeholders, and revisions of the policies need to form an integral part of the decentralized governance model of the future. In essence, governance in decentralized intelligent ecosystems should engender trust, accountability, inclusion, resilience, and alignment with social goals.

18.11 CONCLUSION

In this chapter, future directions for blockchain-based intelligent ecosystems have been addressed based on the integration between decentralized Artificial General Intelligence (AGI), blockchain technology, quantum-resistant cybersecurity and ethical governance. The chapter started with an examination of the development of intelligent computing systems from centralized systems to decentralized ones where autonomous agents interact using secure distributed computing networks. In this chapter, the idea of decentralized AGI has been presented as an alternative to traditional centralized intelligence systems. In contrast to traditional intelligence systems in which all computations take place inside one computational system, decentralized AGI involves intelligent computing ecosystems consisting of autonomous agents collaborating through blockchain technology. The significance of blockchain as underlying infrastructure for distributed intelligence was considered thoroughly. Distributed consensus, decentralized identity management, immutable audit logs, smart contracts,

incentivizing tokenization, and secure provenance of knowledge make it possible to achieve trustful collaboration between independent computational entities. All those functions suggest blockchain technology as one of the key enablers of intelligent systems in the future. Furthermore, the impact of quantum computing on blockchain security was considered. The expected capabilities of quantum computers suggest that transition to post-quantum cryptography that would ensure secure communications, identities, and decentralized governance is necessary. It was shown that hybrid approach to transition, cryptographic standardization, and quantum-resistant blockchains architecture constitute the key components of future cybersecurity framework. Ethics took the leading place in all discussions. Such concepts as transparency, fairness, accountability, privacy protection, sustainability, democratic governance, and human-centric artificial intelligence were proposed as key values that should underlie the development of intelligent decentralized systems. Instead of being an outside regulatory mechanism, ethical reasoning should be built into the architecture and governance of future systems. Specifically, a case study was presented, showing how the development of blockchain-powered decentralized AGI would revolutionize scientific cooperation at a global scale through secure identity management, research provenance transparency, automation of governance, and autonomous research assistants. The presented case study highlighted both the possibilities and challenges of deploying decentralized intelligent systems. In addition, the research directions in decentralized intelligent ecosystems were listed and discussed. They included decentralized foundation models, autonomous governance, edge intelligence, neuromorphic computing, privacy-preserving AI, interoperable blockchain ecosystems, and interdisciplinary governance frameworks. These directions signal that in the future intelligent ecosystems will develop as decentralized computational societies that collaborate and operate in a resilient manner and with full transparency. The integration of blockchain technology, AGI, quantum-proof cryptography, and ethical governance is one of the main technological frontiers of the coming decades. With responsible design and appropriate governance frameworks, decentralized intelligent ecosystems can turn into trustworthy infrastructures of the future digital civilization.

REFERENCES

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
2. M. Swan, Blockchain: Blueprint for a New Economy. O'Reilly Media, 2015.
3. M. Antonopoulos, Mastering Bitcoin, 3rd ed. O'Reilly Media, 2023.
4. M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Wiley, 2009.
5. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.
6. N. Bostrom, Superintelligence: Paths, Dangers, Strategies. Oxford University Press, 2014.
7. S. Russell, Human Compatible: Artificial Intelligence and the Problem of Control. Viking, 2019.
8. Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
9. L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382–401, 1982.
10. National Institute of Standards and Technology (NIST), Post-Quantum Cryptography Standardization Project, Gaithersburg, MD, USA, 2024.
11. G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Foundation, Yellow Paper, 2023.
12. K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016.
13. M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2016.
14. OpenAI, "GPT-4 Technical Report," 2023.
15. World Economic Forum, Global AI Governance Alliance Report, 2024.
16. Garg A, Varshney S, K A, Kansra P. Role of blockchain technology in boosting Ayushman Bharat scheme implementation in India. Indian J Med Res. 2024 May;159(5):449-453. doi: 10.25259/IJMR_594_23. PMID: 39382414; PMCID: PMC11463249.
17. Garg, S. Varshney, A. K., and P. Kansra, "Role of blockchain technology in boosting Ayushman Bharat scheme implementation in India," The Indian Journal of Medical Research, vol. 159, 2024, doi: 10.25259/IJMR_594_23.
18. M. Conti, R. Dehghant says, and S. Kumar, "A Survey of Blockchain Security Issues and Challenges," IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1249-1272, 2021.
19. K. M. K. A. Rahman, M. M. Islam, and M. S. Islam, "Blockchain for Secure IoT: A Review," IEEE Internet of Things Journal, vol. 8, no. 9, pp. 6398-6412, 2021.
20. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Decentralized Business Review, 2008. Available: https://bitcoin.org/bitcoin.pdf
21. Y. Zhang, X. Chen, and B. Wang, "Blockchain-Enabled Secure and Transparent Medical Data Sharing," IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2553-2562, 2020.

A. M. J. S. Islam, M. M. Islam, and M. S. Islam, "Blockchain in Healthcare: Challenges and Opportunities," IEEE Access, vol. 8, pp. 200693-200711, 2020.

22. M. S. Islam, M. M. Islam, and M. S. Islam, "Blockchain-Enabled Secure Data Sharing in Healthcare," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 1, pp. 248-262, 2021.
23. T. M. Nguyen, M. S. Islam, and M. S. Islam, "A Blockchain-Based Framework for Secure Cybersecurity Data Sharing," IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1073-1083, 2021.
24. L. Liu, X. Li, and J. Zhang, "Blockchain for Cybersecurity: A Systematic Review," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 11, pp. 6723-6736, 2021.

25. IEEE Standards Association, "IEEE Standard for Blockchain and Distributed Ledger Technologies," IEEE Std 2418-2020, 2020.
26. K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292-2303, 2016.
27. J. Zhang, H. Wang, and S. Zhang, "A Survey on Blockchain Security and Privacy," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 866-883, 2021.
28. S. M. R. Islam, M. M. Islam, and M. S. Islam, "Secure Healthcare Data Sharing Using Blockchain and Smart Contracts," IEEE Transactions on Cloud Computing, vol. 9, no. 4, pp. 1504-1517,2021.
29. Y. Zhou, K. Zhang, and J. Liu, "Blockchain-Based Secure Data Sharing for IoT in Smart Cities," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3681-3690, 2021.
30. M. M. Islam, M. S. Islam, and M. M. Islam, "Blockchain for Securing Critical Infrastructure and Cyber-Physical Systems," IEEE Transactions on Smart Grid, vol. 12, no. 4, pp. 3417-3428, 2021.
31. F. Li, H. Wang, and Y. Zhang, "A Blockchain-Based Framework for Secure and Privacy-Preserving Medical Data Sharing," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 3, pp. 804-814, 2021.
32. Wang, F. Ye, and X. Zhang, "Blockchain-Enabled Privacy-Preserving Identity Management in Cybersecurity," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2724-2736, 2021.
33. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.
34. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.
35. Dwork, C. (2006). Differential privacy. *International Colloquium on Automata, Languages, and Programming*.
36. Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*.
37. Goldwasser, S., Micali, S., & Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*.
38. Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *IEEE Symposium on Security and Privacy (SP)*.
39. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*.
40. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*.
41. Regulation, G. D. P. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Union*.
42. Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*.
43. W3C Core Working Group. (2022). Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representations. *W3C Recommendation*.
44. Yao, A. C. C. (1986). How to generate and exchange secrets. *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (FOCS)*.
45. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). *Explaining and Harnessing Adversarial Examples*. arXiv preprint arXiv:1412.6572.
46. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Self-published Whitepaper.
47. Wood, G. (2014). *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. Ethereum Project Yellow Paper.
48. Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. P. (2018). *Towards the Science of Security and Privacy in Machine Learning*. IEEE Security & Privacy, 16(1), 11-23.
49. Zhang, R., Xue, R., & Liu, L. (2019). *Security and Privacy in Block chain Systems: A Technical Review*. IEEE Internet of Things Journal, 6(5), 7724-7746.

50. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., & Juels, A. (2020). *Flash Boys 2.0: Frontrunning in Decentralised Exchanges, Miner Extractable Value, and Consensus Instability*. Proceedings of the IEEE Symposium on Security and Privacy (SP), 910-927.
51. Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). *Membership Inference Attacks Against Machine Learning Models*. IEEE Symposium on Security and Privacy (SP), 3-18.
52. Buterin, V. (2014). *A Next-Generation Smart Contract and Decentralized Application Platform*. Ethereum White Paper.
53. Luu, L., Chu, D.-H., Olickel, H., Saxena, P., & Hobor, A. (2016). *Making Smart Contracts Smarter*. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), 254–269.
54. Atzei, N., Bartoletti, M., & Cimoli, T. (2017). *A Survey of Attacks on Ethereum Smart Contracts (SoK)*. *Principles of Security and Trust*, Lecture Notes in Computer Science (LNCS), Vol. 10204, Springer, 164–186.
55. Eyal, I., & Sirer, E. G. (2014). *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*. In *Financial Cryptography and Data Security*. Lecture Notes in Computer Science (LNCS), Vol. 8437, Springer, 436–454.
56. Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). *Eclipse Attacks on Bitcoin's Peer-to-Peer Network*. Proceedings of the 24th USENIX Security Symposium, 129–144.
57. Szabo, N. (1997). *Formalizing and Securing Relationships on Public Networks*. *First Monday*, 2(9).
58. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., & Zanella-Béguelin, S. (2016). *Formal Verification of Smart Contracts: Short Paper*. Proceedings of the ACM Workshop on Programming Languages and Analysis for Security (PLAS), 91–96.
59. Buterin, V., & Griffith, V. (2017). *Casper the Friendly Finality Gadget*. Ethereum Research.
60. Ben-Sasson, E., Bentov, I., Horesh, Y., & Riabzev, M. (2018). Scalable, transparent, and post-quantum secure computational integrity. IACR Cryptology ePrint Archive, 2018, 046. https://eprint.iacr.org/2018/046
61. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., & Juels, A. (2020). Flash Boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 910–927). IEEE. https://doi.org/10.1109/SP40000.2020.00040
62. Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
63. National Institute of Standards and Technology. (2024). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM). https://doi.org/10.6028/NIST.FIPS.203
64. National Institute of Standards and Technology. (2024). FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA). https://doi.org/10.6028/NIST.FIPS.204
65. NIST. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). https://doi.org/10.6028/NIST.AI.100-1
