Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

European Master in Software Engineering

Master Thesis

Blockchain Technologies in Banking – A Framework for Implementation and Analysis

Author: Xue Cheng

Tutor: Dr. Miguel Jimenez Gañán

June, 2025

This Master Thesis has been deposited in ETSI Informáticos de la Universidad Politécnica de Madrid.

Master Thesis

European Master in Software Engineering

Title: Blockchain Technologies in Banking – A Framework for Implementation and Analysis

June / 2024

Author: Xue Cheng

Tutor: Dr. Miguel Jimenez Gañán
Director of the Department of Computer Languages
and Systems and Software Engineering
Universidad Politécnica de Madrid

Abstract

The integration of blockchain technology into the banking sector has the potential to reshape financial infrastructures by enhancing transparency, reducing operational costs, and improving transaction security. This thesis explores the application potential and technical challenges of blockchain in banking from both technical and business perspectives. The research begins with a comprehensive review of the current state of blockchain, including its core principles such as consensus mechanisms, distributed ledger architecture, and data management. Through analysis of pain points in traditional banking systems, the study identifies key areas where blockchain can add value.

A practical implementation framework is proposed, focusing on platform selection, consensus optimization, network topology, data handling, and system maintenance. To validate the framework, a Proof of Concept (PoC) system is designed and implemented using Hyperledger Fabric. The PoC simulates interbank payments between two institutions within a permissioned blockchain network, enabling smart contract-based transaction execution and performance testing.

The results demonstrate the feasibility of blockchain-based interbank solutions and highlight both the advantages and limitations of current technologies. This research contributes technical insights and implementation strategies for banking institutions considering blockchain adoption, and provides a foundation for further academic and industrial exploration.

Table of Contents

1 Introduction

This case study explores the application potential and technical challenges of blockchain technology in the banking industry, with a focus on how it can be integrated into existing financial service infrastructures. As financial technologies evolve, banking institutions have shown increasing interest in decentralized, tamper-proof, and transparent systems, particularly in areas such as payment settlements, data sharing, and process automation. This research responds to that trend by designing and implementing a prototype interbank payment system based on Hyperledger Fabric, aiming to validate the feasibility and implementation strategies of blockchain in real-world banking scenarios.

Based on an initial analysis of bottlenecks and technological needs in current banking systems, the study identifies three core objectives: (1) to evaluate the compatibility of blockchain with banking operations from both technical and business perspectives; (2) to propose a practical implementation framework for blockchain deployment in the banking sector; and (3) to construct a prototype system for empirical validation. The significance of this research lies in its potential to provide guidance on platform selection, system architecture optimization, data management strategies, and system maintenance, while also offering theoretical support for industry standardization and policymaking.

In terms of methodology, this study adopts a combination of literature review and theoretical modeling. First, it reviews relevant national and international research to outline the core principles and mainstream platforms of blockchain technology [1][2][5][6]. Then, it analyzes common issues in traditional banking systems to identify key areas where blockchain could add value. Finally, a banking-oriented implementation framework is proposed, and a prototype system is developed and tested to assess its real-world viability.

This chapter introduces the study through four essential dimensions: the background and motivation behind the research, its objectives and significance, the methodology and approach adopted, and the overall structure of the thesis.

1.1 Research Background and Motivation

To provide a better background, I would mention the uses of BlockChain related to banking and economy regarding cryptocurrencies and DeFi on the one side, and regarding the implementation of applications that benefit from the inmutability and other features of distributed ledgers on the other. This gives a broader view of the background of the work.

1.2 Research Objectives and Significance

This research aims to explore the application potential and technical challenges of blockchain in the banking industry from both technical and business perspectives, and to develop a practical implementation framework. By analyzing the core principles of blockchain—including consensus mechanisms, network architecture, and data management—and aligning them with real-world banking scenarios, the study seeks to identify effective strategies for deployment. The significance of this research lies in the following aspects:

Providing technical guidance and architectural insights for banks adopting blockchain; Supporting the innovation and transformation of banking business models; Enhancing understanding of blockchain system design and maintenance; Contributing theoretical support for policy formulation and industry standardization.

1.3 Research Methodology and Approach

This study adopts a combined approach of literature review and theoretical analysis. First, relevant domestic and international literature is reviewed to systematically outline the foundational concepts, structures, and mechanisms of blockchain technology. Then, based on an analysis of current pain points and trends in the banking industry, the potential value and implementation challenges of blockchain are examined. Finally, drawing from the technical understanding of blockchain, a banking-oriented implementation framework is proposed, addressing platform selection, consensus mechanism optimization, network design, data handling, and system maintenance.

1.4 Thesis Structure

To make the research easier to follow, this thesis is divided into five main chapters: Chapter 1 sets the stage by introducing the topic, explaining why it matters, and outlining the goals and approach. Chapter 2 covers the basics of blockchain—what it is, how it works, and what platforms exist. Chapter 3 looks at how blockchain could change banking, as well as the problems that might come up. Chapter 4 lays out a step-by-step framework for how banks might actually put blockchain into use. Chapter 5 Designed and implemented a Proof of Concept (PoC) system based on Hyperledger Fabric. This experimental system focuses on interbank payments, simulating cross-institutional fund transfers between two banks. By building a permissioned blockchain network, deploying smart contracts, and conducting performance testing, the project explores the practical capabilities and limitations of blockchain technology in financial scenarios. Chapter 6 wraps things up and points out what could be explored next.

2 Overview of Blockchain Technology

2.1 Basic Concepts of Blockchain

This chapter elaborates on the definition of blockchain by analyzing its fundamental aspects, including its underlying data structure, distributed ledger technology, and consensus mechanisms.

2.1.1 Definition of Blockchain

Blockchain is a special type of distributed database designed for secure and transparent data storage [5]. It is decentralized, operates through a consensus mechanism, and ensures that data, once written, cannot be altered. A fundamental feature of blockchain is that it removes the need for trust between participants in the system. Through the consensus mechanism, blockchain enables a trusted and tamper-proof ledger to be maintained among mutually untrusting peers. For example, consider three competing banks, A, B, and C, which do not fully trust each other. Traditionally, they would rely on a trusted third party to coordinate and verify transactions. In a blockchain system, however, each bank holds a complete copy of the ledger, and every transaction must be validated through a consensus algorithm. If any bank attempts to cheat by altering the records, it would need to convince the majority of other banks, which is practically impossible. This design naturally prevents fraud and ensures the integrity of the system.

2.1.2 Structure of Blockchain

The data structure of a blockchain can be viewed as a special type of linked list, but it differs from a regular linked list in two key ways:

- **Hash Pointers Replace Regular Pointers:** In a blockchain, each block is linked to the previous block using a hash pointer instead of a regular pointer. A hash pointer not only contains the address of the previous block but also includes the hash value of the previous block's data, ensuring data integrity and tamper resistance.
- **Data Modification Triggers a Domino Effect:** In a blockchain, modifying the data of any block will change its hash value, which in turn affects the hash values of all subsequent blocks, creating a domino effect. In contrast, modifying an element in a regular linked list does not impact other elements.

The basic data structure diagram of a blockchain is as follows:

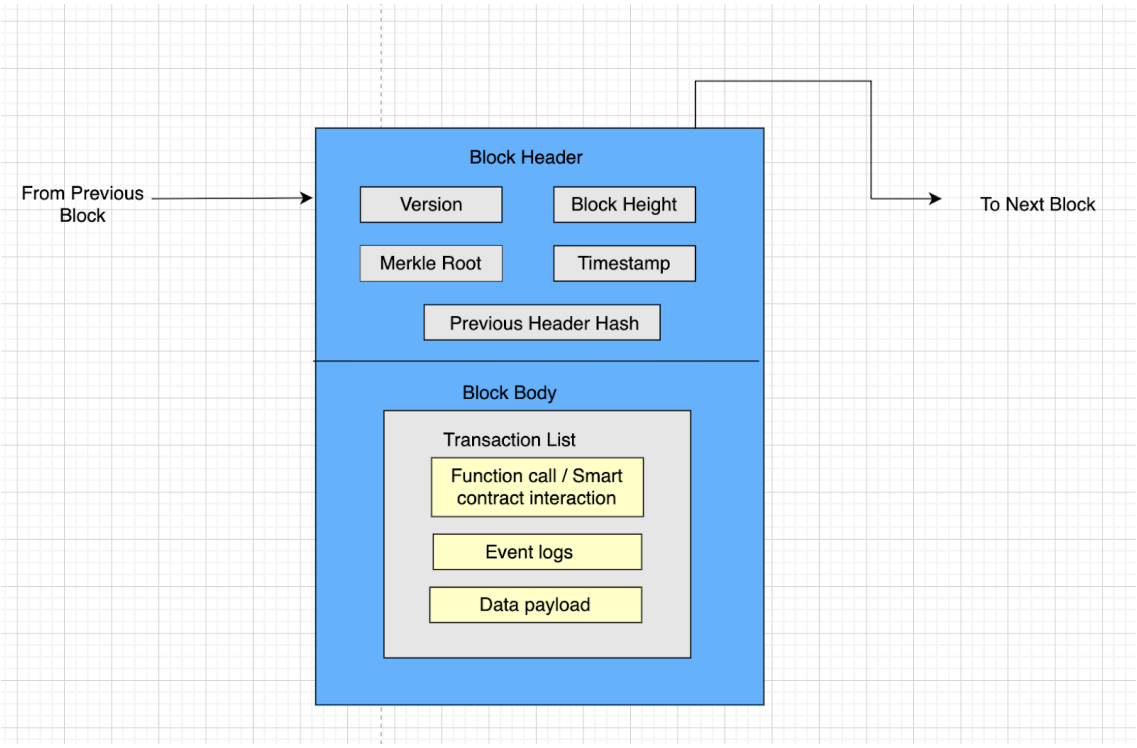

Fig.1.Block structure in a blockchain

The structure of a blockchain consists of a block header and a block body. Block Header includes the following key information:

- **Version**: The version number of the block, used to distinguish different protocol rules.
- **Nonce**: A random number used for Proof of Work (PoW) calculations.
- **Block Height:** The position of the block in the chain, starting from the genesis block (height 0).
- **Difficulty Target**: The target value for adjusting mining difficulty, controlling the complexity of PoW calculations.
- **Merkle Root:** The hash value of all transaction data in the block, used to verify transaction integrity.
- **Timestamp**: The time when the block was created.
- **Previous Block Hash:** The hash value of the previous block, used to link blocks together.

The information in the block header is hashed using a hash algorithm to generate a unique hash value, which is passed to the next block. Notably, the Merkle Root in the header is a field based on the Merkle Tree, a data structure similar to a binary tree but with a key difference: it uses hash pointers instead of regular pointers to link nodes. The data structure diagram of a Merkle Tree(Merkle Proof) is as follows:

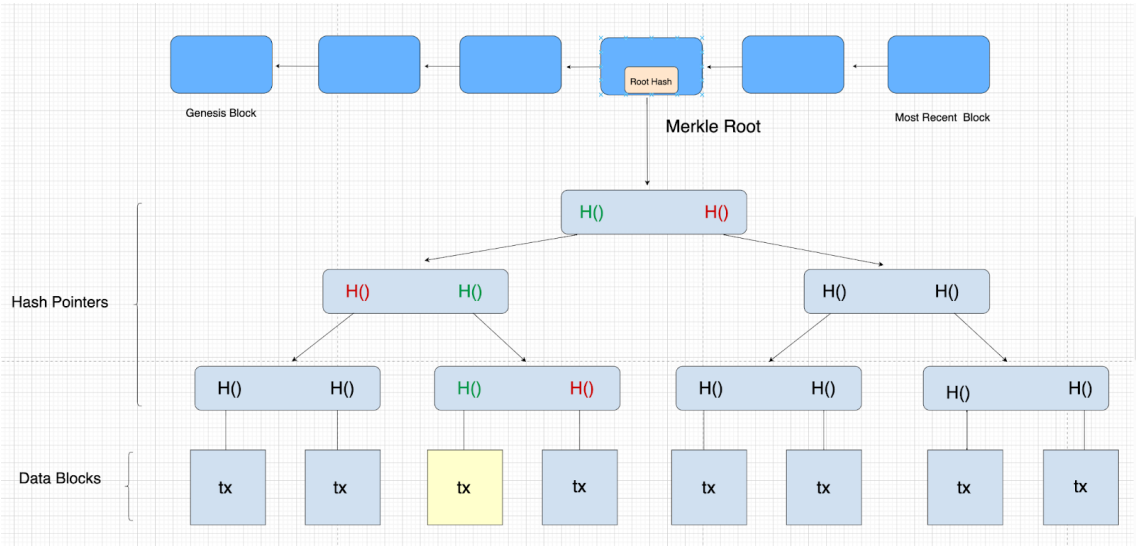

Fig.2. Merkle Tree(Merkle Proof) structure in a blockchain

To determine whether the transaction tx in the yellow area is included in the data chain, we can calculate its hash value using the data from the yellow area and then combine it with adjacent hash values to compute the next-level hash. By repeating this process layer by layer, we ultimately obtain the Root Hash, which allows us to verify whether the transaction is part of the data chain.

Block Body stores the actual transaction data. In cryptocurrency blockchains (e.g., Bitcoin, Ethereum), this typically includes the sender's address, receiver's address, transaction amount, and other financial details. However, in distributed ledger platforms like Hyperledger Fabric, which are designed for enterprise and decentralized applications (DApps), transactions may involve smart contract executions, asset transfers, or business logic operations rather than just cryptocurrency transfers.

2.1.3 Distributed Ledger Technology (DLT)

The core of blockchain is its **distributed ledger**, which is a **decentralized data** storage structure. Unlike centralized databases, a blockchain ledger has no single authority; instead, it is maintained by multiple nodes across the network. Through a **consensus mechanism**, these nodes work together to ensure data consistency and security.

In the previous section (as shown in the blue area of Figure 2), data is stored in blocks and linked together in a chain structure, forming an **immutable ledger**. In section 2.1.2, we mentioned that each block typically contains a Merkle tree to store all transaction records. Nodes that store the **entire blockchain data** are called full nodes, while those that store only the **block headers** are referred to as **light nodes**. Below is a comparison of Full Nodes vs. Light Nodes:

Feature	Full Nodes	Light Nodes
Data Storage	Store the entire blockchain, including all blocks and transactions	Store only headers
Transaction Verification	Can independently verify all transactions	Requires requesting data from full nodes for transaction verification
Computing Resources	High	Low
Use Case	Used by miners, validators, and developers for full transaction history	Suitable for mobile wallets and applications that need quick access with minimal storage

Table.1. Full Nodes vs. Light Nodes

Full nodes ensure the integrity and security of the blockchain, while light nodes offer a more lightweight access method, making them suitable for resource-constrained devices.

In addition to the full node and light node roles in blockchain systems, various Distributed Ledger Technologies (DLTs) employ analogous role divisions, albeit with distinct terminologies and implementation-specific responsibilities. For instance:

- **Full-node-like roles** (e.g., Fabric peers, PoS validators) typically assume critical tasks such as transaction execution, block validation, or ledger replication, requiring substantial computational and storage resources.
- **Hyperledger Fabric** utilizes orderer nodes for transaction ordering (consensus) and peer nodes for ledger storage and transaction validation, reflecting a functional separation similar to full nodes in blockchain.
- **Corda** introduces notary nodes to prevent double-spending (a role distinct from miners in PoW systems), while its vault nodes store only relevant transaction data, mirroring the lightweight behavior of light nodes.
- **PoS-based chains** (e.g., Ethereum 2.0) replace miners with validator nodes, which must stake collateral to participate in block production, whereas light clients (e.g., in Ethereum's LES protocol) prioritize resource efficiency by relying on full nodes for data retrieval.

DLT type	Full node analog role	Light node analog role	Special role
Bitcoin	Full Node	SPV Node	Miner(PoW)
Ethereum2.0(PoS)	Validator Node	Light Client	Beacon Node
Hyperledger	Peer	Gateway Client	Orderer
Corda	Vault Node	Observer Node	Notary

Table.2. Comparison of Consensus Mechanisms

These role variations are inherently tied to the underlying consensus mechanisms (e.g., BFT in Fabric vs. PoW/PoS in public blockchains) and the DLT's design goals (e.g., scalability in IOTA vs. auditability in Corda). A comparative analysis reveals that:

- **Full-node-like roles** (e.g., Fabric peers, PoS validators) typically assume critical tasks such as transaction execution, block validation, or ledger replication, requiring substantial computational and storage resources.
- **Light-node-like roles** (e.g., Fabric gateway clients, Corda observers) trade off autonomy for efficiency, often depending on trusted full nodes for data integrity verification.

2.1.4 Consensus Mechanisms

The consensus mechanism is one of the core technologies of blockchain decentralization. It ensures that all nodes in the blockchain network reach an agreement on the ledger. Without centralized management, each node needs to verify the legitimacy of transactions and determine the addition of new blocks through the consensus mechanism.

2.1.4.1 PoW

Proof of Work (PoW) is a consensus mechanism used in cryptocurrencies like Bitcoin. Miners package new transactions into a block and continuously attempt different nonce(Nonce, as shown in Figure 2 of Section 2.1.2) values until they find one that makes the block's hash meet a specific difficulty condition. Once a valid nonce is found, the block is considered valid, and the miner gains the right to record the block, broadcasting it to the network. Other nodes verify the block's validity and then add it to their respective copies of the blockchain, thereby achieving consensus across the network and confirming the latest state of the blockchain [5][6].

This mechanism operates through the following computationally secured process:

- **Block Construction and Hashing**
  Miners package new transactions into a block and continuously attempt different nonce (as shown in Figure 2 of Section 2.1.2) values. They repeatedly hash the block header (which includes the nonce, previous block hash, and transaction data) using the SHA-256 algorithm until finding a hash output that meets a specific difficulty target.

- **Computational Difficulty**
  The difficulty requirement typically mandates that the block's hash must be below a certain target value, often represented by a number of leading zeros (e.g., 000000000abc123...). This is computationally challenging because:

    1. SHA-256 produces pseudorandom outputs with no predictable pattern.
    2. The only way to find a valid nonce is through brute-force trial and error.

3. The probability of finding a valid hash is extremely low (on the order of 1 in trillions).
4. Modern miners must perform quadrillions of hash calculations per second (PH/s) to be competitive.

- **Difficulty Adjustment Mechanism**

  Bitcoin's network automatically adjusts the mining difficulty every 2016 blocks (approximately two weeks) to maintain a consistent block time of about 10 minutes. The adjustment works as:

  1. If blocks are being mined too quickly (indicating increased network hashrate), the difficulty increases
  2. If blocks are being mined too slowly (indicating decreased hashrate), the difficulty decreases
  3. The adjustment formula: New Difficulty = Old Difficulty × (Actual Time of Last 2016 Blocks / 20160 minutes)

2.1.4.2 PoS (Proof of Stake)

Proof of Stake (PoS) requires participating nodes to demonstrate ownership of a certain amount of tokens in order to gain the right to propose or validate new blocks. This mechanism significantly reduces the energy consumption compared to Proof of Work, making it more sustainable for long-term operation. However, one major concern with PoS is its potential to foster wealth-based centralization. Since nodes with larger token holdings have a higher probability of being selected, this can lead to a situation where wealthier participants consistently dominate block production. Over time, this dynamic may concentrate control in the hands of a few entities, undermining the decentralization and fairness that blockchain systems aim to achieve. This issue is sometimes referred to as the "rich get richer" problem in consensus design, and has prompted ongoing research into hybrid or modified PoS models that introduce randomness, delegation, or identity-based balancing mechanisms to mitigate this risk [5][6].

2.1.4.3 BFT(Byzantine Fault Tolerance)

Byzantine Fault Tolerance (BFT) is a consensus mechanism designed to address malicious nodes, ensuring the consistency and reliability of the system even when some nodes exhibit faults or malicious behavior. It usually supports a maximum of malicious nodes with respect to the total [5][6].

Core Theorem

- Total network nodes: n
- Maximum tolerable faulty nodes: $f \leq (n-1)/3$

he system requires at least 3f+1 nodes to tolerate f malicious nodes, for example: Example: A 4-node system can tolerate at most 1 malicious node ($4 \geq 3\times1 +1$),The system may fail if more than 1/3 of nodes become malicious.

2.1.4.4 PBFT(Practical Byzantine Fault Tolerance)

Practical Byzantine Fault Tolerance (PBFT) is a well-known practical implementation of the theoretical Byzantine Fault Tolerance (BFT) model. While BFT defines what is theoretically achievable in terms of reaching consensus in the presence of Byzantine faults, PBFT provides a concrete mechanism for how to achieve it. Their relationship can be compared to that of an interface and its implementation: BFT specifies the capability, while PBFT delivers the method [5][6].

PBFT is characterized by its efficiency and practicality. It employs a three-phase consensus protocol—pre-prepare, prepare, and commit—making it suitable for asynchronous network environments, where message delays are unpredictable:

1. **Pre-prepare**: The primary (leader) node receives a client request and broadcasts a proposal to other nodes.
2. **Prepare**: Upon receiving the proposal, nodes exchange confirmation messages, signaling that they have received and agree with the proposal.
3. **Commit**: After collecting a sufficient number of prepare messages (at least 2f+1), nodes broadcast a commit message and, once the threshold is met, execute the request.

Compared to early BFT algorithms, PBFT features a simpler design and offers better performance in fault-free environments. When there is no failure, only $O(n^2)$ messages are needed (n is the number of nodes), which is more efficient. Moreover, it does not rely on tokens or mining, making it well-suited for deployment in permissioned blockchains, consortium networks, and financial systems where high security and strong consistency are required.

2.1.4.5 DPoS (Delegated Proof of Stake)

In the DPoS mechanism adopted by BitShares, token holders vote to elect a group of witnesses. Each witness takes turns to receive a 2-second time window to produce a block. If a witness fails to generate a block within the allotted time, the permission is automatically passed to the next witness in the sequence. Additionally, token holders can vote to replace witnesses at any time, ensuring the network's decentralization and efficient operation.

Delegated Proof of Stake (DPoS) is a consensus mechanism that improves on traditional Proof of Stake (PoS) by solving issues like centralization, low participation, and the "nothing at stake" problem. Instead of relying solely on large token holders to validate blocks, DPoS allows all token holders to vote for a small group of delegates who take on the task of block production and validation. These delegates can be replaced at any time through voting, which increases accountability and decentralization. With fewer active participants, DPoS also achieves faster transactions and greater efficiency. In addition, it lowers the technical barriers to participation, encouraging wider community involvement while keeping the network secure.

Different consensus mechanisms have their own advantages and disadvantages:

Consensus Mechanism	Advantages	Disadvantages
PoW	• Highly decentralized, allowing any node to participate • High security, preventing malicious attacks and tampering	• High consumption, requiring a large amount of computing resources • Higher network latency, resulting in slower processing speeds
PoS	• Energy-efficient, consuming fewer computing resources than PoW • Attackers need to control a majority of the network's stake to launch an attack, enhancing security.	• May lead to a "rich get richer" effect, reducing the level of decentralization. • Requires effective mechanisms to prevent centralization risks.
BFT	• High efficiency, capable of handling a high transaction throughput. • Does not require energy consumption like PoW and PoS.	• Performance may degrade as the network scales. • More complex deployment, requiring nodes to have a high level of trust.
DPoS	• Achieves rapid consensus within seconds, ensuring consistency.	• Less decentralized compared to public blockchain consensus mechanisms. • More suitable for multi-center business models with known participants.

Table.3. Comparison of Consensus Mechanisms

2.1.5 Smart contracts

Smart contracts are self-executing programs stored on the blockchain that automatically enforce and execute predefined rules when certain conditions are met. They play a central role in many blockchain platforms, enabling decentralized applications (dApps) and automating transaction validation beyond simple value transfers. Unlike traditional contracts, smart contracts eliminate the need for intermediaries and operate transparently on distributed

networks. On platforms like Ethereum, smart contracts are essential for creating decentralized services, while in enterprise-focused blockchains such as Hyperledger Fabric, they enable complex business logic within a permissioned environment. For instance, in trade finance and letter of credit (LoC) management, smart contracts can significantly reduce reliance on paper documents, streamline cross-border workflows, and improve processing efficiency by automating the execution of predefined terms between banks, exporters, and importers.

2.2 Network Architecture (Open vs. Permissioned Networks)

This section provides a comprehensive description of Open and Permissioned Blockchain Networks, examining their structural and functional features, followed by a comparative analysis.

2.2.1 Open Networks

A public chain is an open blockchain where anyone can read data, send transactions, and participate in the consensus process, with transactions being effectively confirmed. Public chains typically employ consensus mechanisms such as **Proof of Work** (PoW), **Proof of Stake** (PoS), or **Delegated Proof of Stake** (DPoS), combining economic incentives with cryptographic verification. They follow the principle of "distribution according to contribution," meaning that the economic rewards participants receive are proportional to their workload. Such blockchains are generally considered fully decentralized.

2.2.2 Permissioned Networks

Permissioned chains, also known as **consortium chains**, are typically blockchains used within specific institutions or organizations, where only consortium members can participate. In such networks, a subset of nodes is pre-selected to handle transaction validation, and block generation is collectively determined by these validator nodes. Other nodes can participate in transactions but do not have validation rights. Both read/write permissions and validation rights are defined according to the consortium's rules. Each node usually corresponds to a real-world entity or organization and must be authorized to join or exit the network. Consortium chains generally employ consensus mechanisms such as **Proof of Stake** (PoS) or **Byzantine Fault Tolerance** (BFT).

The characteristics of these two different network architectures are as follows:

Feature	Open	Permissioned
Participation	Unrestricted, anyone can join	Authorized, only specific nodes can join
Decentralization	Fully decentralized	Partially decentralized or centralized
Transparency	Fully transparent	Limited transparency
Consensus Mechanism	PoW, PoS, etc	BFT, Raft, etc
Use Cases	Cryptocurrencies, etc	Enterprise applications

Table.4. Feature of Network

2.3 Blockchain Platforms (Hyperledger Fabric, Corda, Ethereum)

These three platforms share common features, including support for smart contracts, the use of distributed ledger technology, suitability for enterprise applications, and being open-source projects with active developer communities and ecosystems. However, each platform also has its unique characteristics. Corda focuses on the financial sector, employing a "need-to-know" data-sharing approach and utilizing notary nodes. Ethereum is a public blockchain platform that is fully decentralized and boasts a vast developer ecosystem. Hyperledger Fabric, on the other hand, is a consortium blockchain platform known for its high customizability and strict access control, making it ideal for enterprise-level applications.

Fig. 3 illustrates the ecosystems of Hyperledger Fabric, Corda, and Ethereum, highlighting their shared characteristics as well as their unique features:

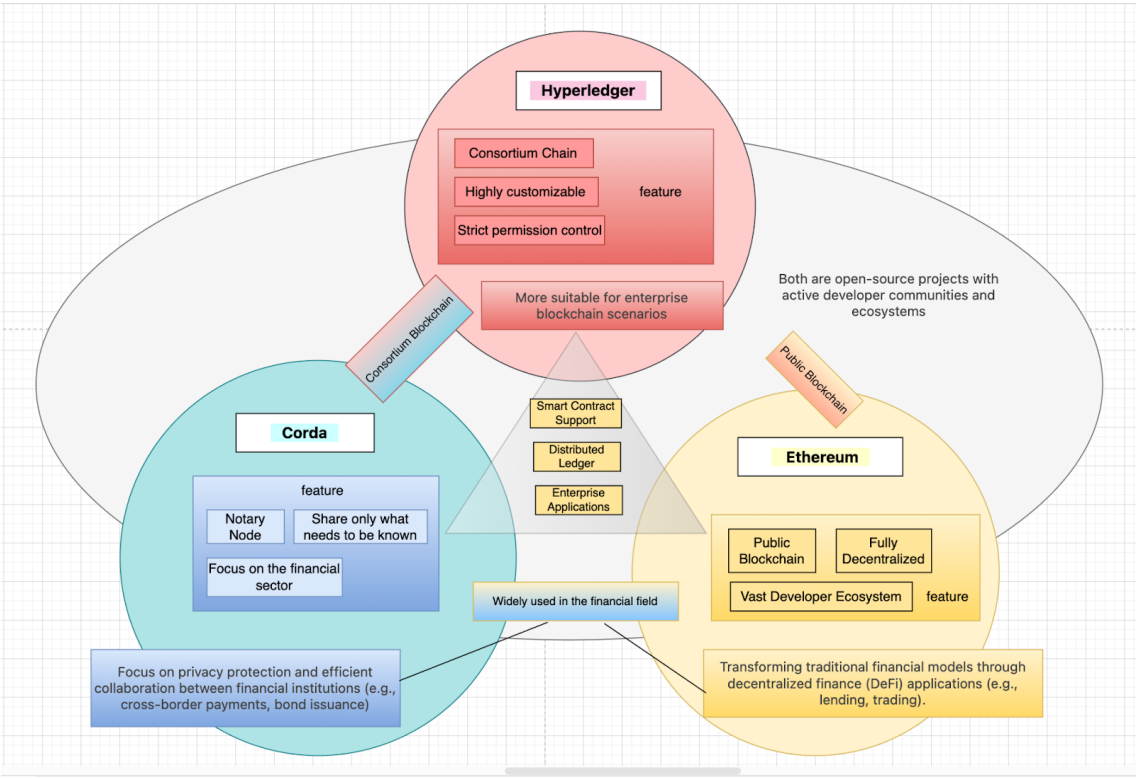

Fig.3. Hyperledger Fabric, Corda and Ethereum ecosystem

Figure 3 provides a comparative analysis of three leading blockchain platforms—Hyperledger Fabric, Ethereum, and Corda—highlighting their architectural distinctions and industry applications. Hyperledger Fabric, as a consortium chain, emphasizes enterprise adaptability with its permissioned structure, customizable features, and strong compliance controls, making it ideal for financial and supply chain solutions. In contrast, Ethereum operates as a public, decentralized network, fostering innovation through smart contracts and decentralized applications (DApps), particularly in decentralized finance (DeFi) and Web3 ecosystems. Meanwhile, Corda specializes in financial-sector efficiency, employing a privacy-focused model where transactions are shared only among relevant parties and validated via notary nodes, optimizing workflows like cross-border payments and securities issuance. While all three platforms support smart contracts and have active open-source communities, their designs cater to different needs: Fabric excels in regulated enterprise environments, Ethereum enables trustless public applications, and Corda streamlines inter-institutional financial operations with minimal data exposure. This comparison underscores the importance of selecting a blockchain framework based on specific use-case requirements, whether prioritizing decentralization (Ethereum), governance (Fabric), or privacy-aware collaboration (Corda).

2.3.1 Hyperledger Fabric

Hyperledger Fabric is an excellent implementation of a consortium blockchain, making it particularly suitable for enterprise blockchain scenarios. It features comprehensive access control, where members must pass identity verification

to join the network, balancing data sharing with privacy protection for enhanced security. It adopts a modular design, with pluggable consensus mechanisms and encryption algorithms. Specifically, Fabric supports several consensus mechanisms such as Solo for testing, Kafka (deprecated) for ordered message queuing, and Raft, which is currently the recommended production-ready consensus protocol. Additionally, there are ongoing efforts to incorporate Byzantine Fault Tolerant (BFT) consensus algorithms.

It utilizes container technology, where nodes and chaincode run in Docker containers, ensuring isolated environments while enabling communication between containers [3][6]. Table 5 is a summary of Hyperledger Fabric advantages, disadvantages, and use cases :

Hyperledger Fabric	
Advantages	• **High Customizability**: Hyperledger is a highly customizable blockchain platform that supports private, consortium, and public networks. • **High Performance and Scalability**: The Hyperledger platform typically offers high performance and scalability, making it suitable for enterprise-level applications. • **Rich Tools and Libraries**: Hyperledger provides a wide range of tools and libraries, supporting the development of complex enterprise applications.
Disadvantages	• **Steep Learning Curve**: Hyperledger has a relatively steep learning curve, requiring a solid understanding of blockchain technology and enterprise applications. • **Smaller Community Size**: Compared to public blockchain platforms like Ethereum, Hyperledger's community size is relatively smaller.
Use Case	• **Enterprise Application Development**: Ideal for developing highly customizable and security-critical enterprise applications. • **Supply Chain Management**: Suitable for building supply chain management systems to achieve traceability and transparency.

Table.5. Hyperledger Fabric Platform Analysis

2.3.2 Corda

Corda is a blockchain platform specifically designed for enterprises and financial institutions. Its core idea is to make collaboration between businesses more efficient and private.Corda's architecture divides the network into distinct "compatibility zones," each acting like an independent circle where different smart contracts can run to meet various business needs. A compatibility zone defines a shared network governance and identity model for a set of nodes that interact under the same business rules and regulatory requirements. Each zone

may operate independently with its own notary services, certificate authorities, and network parameters [3][6].

Corda's standout feature is "sharing only what needs to be known." In traditional blockchains, all nodes store the entire dataset, but in Corda, each node only stores transaction data relevant to itself, completely hiding unrelated information. This approach protects privacy while reducing data storage pressure. Table 6 summarizes the advantages, disadvantages, and use cases of Corda:

Corda	
Advantages	• **Privacy and Confidentiality:** Corda is an enterprise-focused blockchain platform that supports private transactions and data privacy protection. • **Smart Contract Support**: Corda supports smart contracts, but unlike Ethereum, Corda's smart contracts can only access data related to the transaction, ensuring transaction privacy. • **Tailored for the Financial Industry**: Corda was initially designed for the financial sector, offering strong support for financial transactions and compliance.
Disadvantages	• **Relatively Closed Ecosystem**: Corda is relatively closed, with a smaller ecosystem and fewer available tools and libraries. • **Steep Learning Curve**: Corda has a relatively steep learning curve, requiring knowledge of blockchain technology and financial operations
Use Case	• **Financial Transactions**: Ideal for developing private transaction systems in the financial industry, such as cross-border payments and bond issuance. • **Data Sharing**: Suitable for secure data sharing and collaboration between enterprises while protecting data privacy.

Table.6 Corda Platform Analysis

2.3.3 Ethereum

Ethereum is an open blockchain platform, somewhat like a "global computer" on the internet. A key feature of Ethereum is that "everyone can participate"—anyone can run a node and join the network. Its transactions and smart contract code are publicly transparent, viewable by all, while encryption ensures security [3][6]. Currently, Ethereum uses the Proof of Work (PoW) mechanism, commonly known as "mining," but it will transition to Proof of Stake (PoS) in the future, making it more environmentally friendly and efficient.

Ethereum successfully completed its transition from Proof-of-Work (PoW) to Proof-of-Stake (PoS) through "The Merge" upgrade in September 2022 [11].

Table 7 provides a summary of Ethereum's advantages, disadvantages, and use cases:

Ethereum	
Advantages	• **Smart Contract Support**: Ethereum supports smart contracts, enabling automated business logic. • **Decentralized Application Development**: Ethereum is a public blockchain platform ideal for developing decentralized applications (DApps). • **Vast Ecosystem**: Ethereum boasts a large developer community and a rich array of third-party tools, libraries, and services.
Disadvantages	• **Performance Issues**: Ethereum struggles with performance when handling large-scale transactions, often leading to network congestion. • **Lack of Privacy:** Ethereum's blockchain is public, lacking privacy, which makes it unsuitable for certain use cases.
Use Case	• **Decentralized Application Development:** Ideal for developing highly decentralized applications, such as decentralized finance (DeFi) applications. • **Digital Asset Trading**: Suitable for digital asset trading and crowdfunding activities.

Table.7 Ethereum Platform Analysis

When choosing a blockchain platform, it is important to weigh your options based on your needs and specific circumstances. If you are a developer looking to build decentralized applications, Ethereum might be a good choice. If you are an enterprise aiming to develop private blockchain applications, then Hyperledger or Corda might be more suitable

3 Prospects and Challenges of Blockchain in Banking

3.1 Prospects of Blockchain in Banking

This section elaborates on the prospects of blockchain in banking by examining four key aspects: enhancing operational efficiency, transforming future business models, the development of Central Bank Digital Currencies (CBDCs), and innovative applications in data and privacy protection.

3.1.1 Enhancing Banking Operational Efficiency

Currently, the operations of banks primarily rely on centrally controlled ledger systems. However, the future may herald a shift towards blockchain and network-centric models, supplanting the traditional centralized management. In some banks, services such as peer-to-peer lending and robo-advisors have already emerged. These services bypass the traditional role of banks as intermediaries, rendering transactions more direct and efficient.As shown in Figure 4:

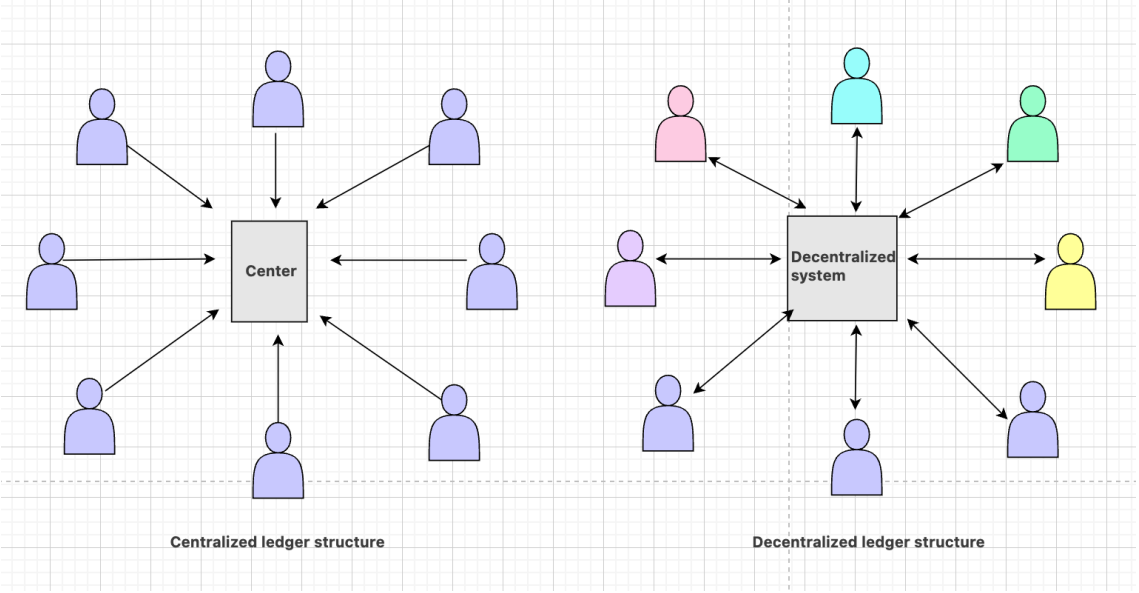

Fig.4.Comparison between Centralization and Decentralization

The essence of blockchain technology lies in its distributed ledger characteristic, which means it does not rely on a single central point. Once data is recorded, it cannot be altered and is transparent to all participants. These features are particularly useful in cross-border payments. Traditional cross-border payments typically require multiple intermediary institutions, resulting in a complex process, high costs, and long transaction times. Blockchain technology, on the other hand, enables direct peer-to-peer transactions, eliminating

intermediaries and significantly improving efficiency. Banks can choose to use public, consortium, or private blockchains as needed.

The traditional banking business model typically adheres to Treacy and Wiersema's value disciplines, which include striving for operational excellence (such as reducing costs, improving reliability and accuracy), enhancing customer intimacy (improving service quality), and strengthening product leadership (launching innovative products and services). With the introduction of blockchain technology, banks can optimize back-end processes by reducing redundancy and accelerating settlements. Moreover, blockchain can enhance the transparency of reporting and the efficiency of data dissemination, thereby strengthening regulation. In addition to improving service quality, it can also help banks reach new customers and enter new markets, such as developing digital asset platforms or international trade.

In summary, the application of blockchain technology in the banking industry is mainly reflected in the following aspects:

- Payments and Cross-border Settlements: Reducing the cost of cross-border payments and speeding up transaction times.
- Smart Contracts: Automating contract execution, reducing intermediary links, and increasing transparency.
- Trade Finance and Letter of Credit Management: Reducing the use of paper documents and improving processing efficiency.

Through these improvements, blockchain technology has the potential to make banking operations more efficient, transparent, and secure.

3.1.2 Transformation of Future Banking Business Models

Traditional banks rely on intermediaries to complete cross-border payments and settlements, whereas the decentralized nature of blockchain, combined with smart contracts and distributed ledger technology, can significantly reduce reliance on intermediaries, lower transaction fees, and increase transaction speed. For instance, Ripple's blockchain network, RippleNet, has been adopted by institutions such as Spain's Santander [7].

Bank, reducing cross-border payment times from several days to mere seconds. JPMorgan Chase's JPM Coin has enabled instant settlements between institutional clients through blockchain. Additionally, Singapore's DBS Bank has launched a digital trading platform that supports security token offerings and digital asset trading, providing customers with new investment opportunities.

In the future, banks can leverage other platforms developed through blockchain technology to expand new financial products and trading models. For example, banks can issue digital assets, such as tokenized real estate or art, lowering investment thresholds and increasing liquidity.

One of the most transformative innovations in this context is Decentralized Finance (DeFi)—a financial ecosystem built on blockchain technologies that enables the creation and operation of financial services, such as lending, borrowing, trading, and asset management, without relying on centralized intermediaries like banks. DeFi applications typically operate on public blockchains like Ethereum and are powered by smart contracts. Key

components of the DeFi ecosystem include tokenized assets, decentralized exchanges (DEXs), and automated market makers (AMMs).

Simultaneously, banks can establish cooperative relationships with DeFi platforms, offering custodial services and integrating DeFi products into existing services to provide customers with more choices. For instance, Standard Chartered Bank has partnered with Metaco to launch an institutional-grade digital asset custody platform, offering custodial services for cryptocurrencies and DeFi assets to institutional clients.

To ensure compliant operations, banks can collaborate with regulatory bodies to ensure that blockchain applications adhere to legal and regulatory standards. For example, when issuing blockchain-based bonds, Société Générale worked closely with EU regulators to ensure compliance. Moreover, banks can establish comprehensive risk management systems to address the volatility and potential risks of the blockchain market.

Blockchain technology not only reduces the costs of cross-border payments and settlements but also significantly enhances efficiency through real-time settlements and automated processes. For customers, this means faster transaction speeds and lower fees. Additionally, distributed ledger technology ensures transaction transparency and traceability, bolstering customer trust. In the future, with the proliferation of Central Bank Digital Currencies (CBDCs) —a topic explored further in Section 3.1.3, banks can develop related services, such as CBDC custody and payments. The application of blockchain technology in green finance is also highly anticipated, such as the tokenization of carbon credits and the issuance of green bonds. Through blockchain, customers can also securely monetize their data, creating new revenue streams.

Beyond payments, blockchain's potential extends to several innovative applications in finance, including:

- **Green Finance**: Blockchain enables the tokenization of carbon credits and the issuance of green bonds. These digital instruments can be transparently tracked on the blockchain, ensuring the authenticity and traceability of environmental claims, which is critical for ESG reporting and compliance.
- **Data Monetization**: Individuals and enterprises can securely monetize their personal or operational data through blockchain-based platforms, maintaining ownership and privacy while enabling controlled access for value exchange.
- **Asset Tokenization**: Real-world assets such as real estate, art, or commodities can be fractionalized and traded on blockchain networks, reducing entry barriers and increasing liquidity.

These applications illustrate how blockchain is not only transforming payments, but also opening new frontiers in sustainable finance, digital asset services, and data-driven business models.

Blockchain technology is leading a revolutionary transformation in the banking industry, from cross-border payments to asset tokenization, and to collaborations with DeFi platforms, the future of banking is brimming with limitless possibilities. Through blockchain, banks can not only improve efficiency and reduce costs but also offer customers more innovative financial services, reshaping the entire financial ecosystem.

3.1.3 Development and Impact of Central Bank Digital Currencies (CBDCs)

Currently, many countries are implementing central bank digital currencies (CBDCs) to adapt to the rapidly evolving digital world. There are two types of digital currencies: Retail and Wholesale.Retail CBDCs are used for payments between individuals and businesses [4]. Wholesale CBDCs are used for interbank settlements. As shown in Figure 5 and 6:

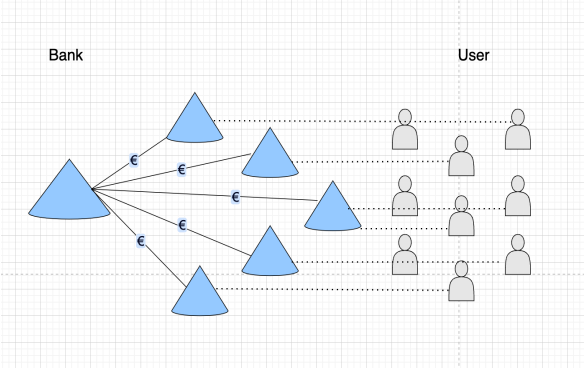

Fig.5.Retail

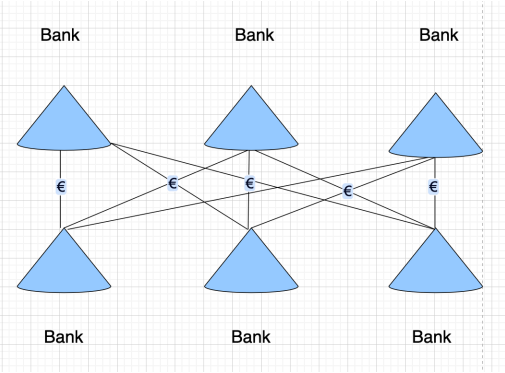

Fig.6.Wholesale

The primary distinction between central bank digital currency (CBDC) and existing traditional currency lies in the fact that traditional currency encompasses physical forms such as banknotes and coins, as well as electronic money in bank accounts [12]. In contrast, CBDC is a legal tender in digital form issued by the central bank. Traditional currency possesses a certain degree of anonymity, whereas CBDC can offer a level of privacy protection. Traditional electronic payments require an internet connection to complete transactions, but CBDC can facilitate completely offline payments, providing convenience for many remote areas without network access. The convenience of offline payment also brings about the problem of double payment. In offline CBDC transactions, the double-spending problem is solved by using secure devices such as smart cards, SIM cards, or mobile devices equipped with security modules [13]. These devices can store digital currency locally and have built-in anti-reuse mechanisms to ensure that funds cannot be spent again once they are used. In addition, offline payments usually have an amount limit, and after the device is reconnected to the network, all transactions will be uploaded and compared with the central system to identify whether there is fraud. Some designs also introduce one-time encrypted tokens, each with a unique digital signature, so that the receiving device can verify its validity and prevent duplicate spending even when used in an offline environment [13].

Moreover, CBDC has "legal tender" status [12]; a simple example is that while supermarkets may refuse to accept traditional card or mobile payments, they cannot refuse cash or CBDC payments, as both are considered equivalent.

As shown in Figure 7: In the existing model: the central bank and commercial banks jointly manage the issuance and circulation of currency. In the central bank digital currency model, the role of commercial banks is partially retained.

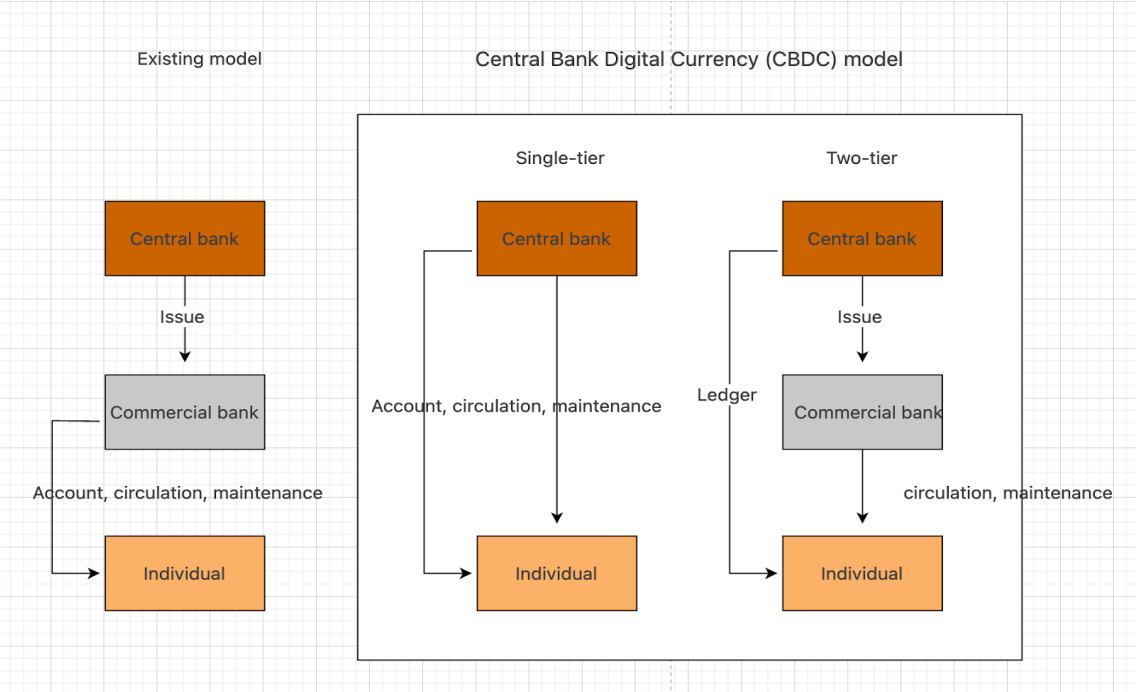

Fig.7.Comparison between the Existing Model and the Central Bank Digital Currency (CBDC) Model

➢ **Existing Model**

The central bank is only responsible for currency issuance, while commercial banks act as key intermediaries, handling account management, currency circulation, and maintenance.

➢ **CBDC Model**

- Single-tier architecture: The central bank interacts directly with the public, without relying on commercial banks [12].
- Two-tier architecture: Commercial banks still participate in circulation and maintenance, but the central bank plays a more dominant role [13].

CBDC strengthens the central bank's direct control and weakens the intermediary function of traditional banks. If CBDCs are widely adopted, the public may shift deposits to CBDC accounts, leading to deposit outflows, reduced lending capacity, and a decline in payment-related business for commercial banks.

While these developments pose clear challenges to the traditional banking sector, it is also important to acknowledge the potential benefits that central bank digital currencies (CBDCs) could bring to the financial system and its users, the benefits brought by digital currency also include:

- It can reduce or even eliminate the need for transfers and payments through financial intermediaries such as commercial banks and payment service providers, helping to reduce friction costs for users [14].
- Currently, some developed countries still have a portion of the population without bank accounts, and commercial banks are often

unwilling to open accounts for individuals without sufficient physical presence, but this issue can be resolved by opening accounts directly with the central bank [15].

While bringing advantages, the challenges that need to be faced are: the current performance of blockchain technology systems is lower than that of traditional centralized systems, so scalability needs to be improved, and banks need to consider user privacy, regulation, security, and other issues.

3.1.4 Innovative Applications in Data and Privacy

In blockchain technology, different encryption algorithms are applied in various scenarios. Hash algorithms, due to their irreversible nature, ensure data integrity and tamper resistance [16]. Asymmetric encryption algorithms are used to generate key pairs (public and private keys), enabling identity verification and digital signatures [17]. Zero-knowledge proofs utilize technologies such as zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge), which can verify the authenticity of data without revealing specific information, thereby enhancing privacy protection [18]. The combination of these encryption algorithms enables blockchain to achieve decentralization, security, and privacy protection. By integrating these technologies, a reliable foundation is also established for the future development of blockchain technology in the banking sector.

3.2 Challenges of Blockchain in Banking

Blockchain technology is a trusted new environment that combines many technical integrations and frameworks. The emergence of new technologies will also bring more business models, but there will still be some challenges.

3.2.1 Technical Challenges

The application of blockchain technology in the financial field is developing rapidly. The existing business needs in the financial field must exceed the performance of blockchain, resulting in low platform availability. If a large-scale attack is encountered, the availability of the platform will be directly affected, and it may even be unable to provide services to users for a period of time.

Additionally, the smart contract in the blockchain can allow the two parties involved in the contract to reach an agreement, without the need for any central agency, and automatically execute through code, reducing the risk of human modification and intervention. However, once a smart contract is released, it is difficult and almost irreversible. If there is a security vulnerability in the contract, the entire chain of projects will be affected, and it is currently difficult to repair, which is also a big guess [19].

Furthermore, the current blockchain technology is not mature enough and the degree of standardization is also low. Banks need to cooperate with industry organizations to jointly formulate standards and promote unified management. At the same time, if banks want to adopt blockchain technology, data storage cannot simply copy the structure of the old database, but needs to be redesigned to adapt to the characteristics of blockchain [20].

3.2.2 Regulatory and Compliance Issues

The application of blockchain technology in the financial industry has brought new challenges to existing laws, regulations and regulatory frameworks. This requires joint research and cooperation among regulatory agencies, legislative departments and financial institutions in various countries. In addition, there is no unified standard system, such as how to design ledgers or application processes, which also requires cooperation with governments and other participants. At the same time, each country's regulatory policies on finance are also different, which further complicates the problem [21].

3.2.3 Cost and Profitability Issues

If banks want to encapsulate their business with blockchain technology, they need to invest a lot of money in hardware, software, and manpower. For example, high-performance servers and storage devices are needed to support the operation of the blockchain network, and they also have to bear the maintenance and upgrade costs of these systems. In addition, banks need to invest in software development, transform legacy systems into blockchain-based systems, and recruit specialized blockchain technology professionals or train existing employees.

A key consideration in this process is whether to deploy blockchain systems on-premise or through cloud-based services. On-premise deployment generally requires substantial upfront investment in physical infrastructure and ongoing costs for system maintenance and IT personnel. In contrast, cloud-based deployment can reduce initial hardware expenditures and offer greater scalability and flexibility, as resources are provided on demand by third-party providers. This approach may also lower the need for in-house IT staff, allowing banks to focus more on core business development and innovation. However, cloud deployment could raise concerns around data security, compliance, and control [22][23]. Therefore, banks must carefully weigh the short-term costs and long-term profitability of each deployment strategy. While both models require significant investment, a well-chosen deployment model may lead to greater operational efficiency and reduced long-term costs.

3.3 Comparison between Blockchain and Traditional Banking Technologies

One of the main differences between blockchain and traditional banking systems is the way transactions are recorded. In traditional banking systems, transactions are recorded by centralized financial institutions such as banks, which store the data in their own databases. Unlike blockchain, these databases are not inherently immutable, so ensuring the reliability and integrity of transaction records requires

To maintain data reliability and prevent tampering or fraud, traditional banking systems rely on multiple layers of security. First, access to sensitive systems and data is tightly controlled through role-based access control (RBAC), multi-

factor authentication, and user authorization protocols. Only authorized personnel can access or modify transaction records, minimizing the risk of internal fraud [24]. Second, all activities involving financial data are logged in detailed audit trails. These logs are regularly reviewed and audited to detect unauthorized access or suspicious behavior. Additionally, encryption is used to protect data both in transit and at rest, and intrusion detection systems are employed to monitor for cyber threats. Banks also comply with strict regulatory standards and implement internal policies to ensure that data handling meets security and privacy requirements.

However, despite these extensive security measures, traditional banking systems still face a non-negligible risk of data tampering. Since transaction data is stored in centralized databases under the full control of the institution, authorized administrators or malicious insiders may, in theory, alter records if security measures are bypassed or fail. This highlights a key limitation of traditional systems compared to blockchain, where immutability is enforced at the protocol level through cryptographic hashing and consensus mechanisms. As a result, while traditional systems rely heavily on external controls to ensure trust, blockchain builds trust into the system itself through its decentralized architecture.

In blockchain, transactions are recorded in a public and decentralized online ledger that is managed by all users in the network. In addition, blockchain uses distributed storage to eliminate single points of failure. Even if some nodes fail, the entire network account can still run continuously.

4 Blockchain Implementation Framework in Banking

This section compares three major enterprise blockchain platforms—Corda, Fabric, and Quorum—highlighting their differences in privacy, smart contract capabilities, and system performance.

4.1 Requirement Analysis for Blockchain Implementation in Banking

Blockchain technology has great potential in the banking industry, mainly because of its unique advantages. At its core, blockchain enables trustworthy operations among mutually untrusted participants by relying on a decentralized consensus mechanism rather than on any single trusted party(Mentioned in section 2.1.1). This means the security and integrity of the system are guaranteed by the protocol itself, not just by the individual players. Through a distributed ledger, transaction data is collectively recorded across multiple nodes, making it extremely difficult to tamper with once confirmed. Additionally, authorized users can transparently view all transactions, which enhances oversight, reduces fraud risks, and strengthens regulatory compliance. These features are further enhanced in permissioned blockchain networks commonly used in banking, where access is restricted to known participants while maintaining the system's inherent trustworthiness.

Another key feature is "smart contracts," which can automatically carry out certain rules. This reduces the need for intermediaries, lowers operational costs, and improves efficiency.

However, blockchain isn't perfect. Its processing speed may not be as fast as traditional banking systems, especially when dealing with a large number of transactions. Also, while transparency is a strength, some transactions involve sensitive information. So, finding a balance between openness and privacy is a challenge. Plus, the decentralized nature of blockchain might not align well with existing financial regulations, which also needs to be addressed [25].

Therefore, when designing blockchain systems, it's important to consider these challenges and align the design with the bank's specific needs—like cross-border payments and trade finance. In the future, as the technology continues to evolve, blockchain could play a bigger role in areas like central bank digital currencies and green finance, bringing new opportunities to the banking industry [1][5][6].

4.2 Selection of Blockchain Platforms

Choosing the right blockchain platform is a critical step for banks when adopting blockchain technology. Different platforms vary significantly in terms of performance, security, and scalability. Selecting the right platform can lead to more efficient system operations, lower development costs, and easier future expansion. On the other hand, choosing the wrong one could cause

various problems. Therefore, banks need to choose the most suitable blockchain platform based on their specific business needs and technical requirements.

When selecting a blockchain platform, banks should focus on the following key factors:

1. **Performance**: One key consideration is whether the platform can handle high transaction throughput and processing speed to meet the demands of banking operations with high concurrency.
2. **Security**: It is important to evaluate whether the platform uses reliable consensus mechanisms and encryption technologies to ensure data security.
3. **Privacy Protection**: A critical factor is whether the platform supports private transactions or data isolation features to meet the banking industry's requirements for customer privacy.
4. **Compliance**: Compliance depends not only on whether the platform supports financial regulations such as Anti-Money Laundering (AML) and Know Your Customer (KYC), but also on the type of blockchain network it adopts. Permissioned networks, which allow only authorized participants, are more suitable for meeting regulatory requirements compared to permissionless networks that are open to anyone.
5. **Development Cost**: It is also worth assessing whether robust development tools are available and whether there is an active developer community to provide technical support during development.
6. **Ecosystem Support**: It is important to consider whether the platform has a mature ecosystem of third-party tools and plugins that can be easily integrated into existing systems.

Mainstream blockchain platforms such as Hyperledger Fabric, Corda, and Ripple will be briefly compared in the following section.

4.2.1 Hyperledger Fabric

Hyperledger Fabric restricts data access to authorized nodes through a robust access control mechanism, ensuring transaction privacy and security. Its modular architecture supports customizable smart contracts (Chaincode), allowing flexible deployment of functional modules based on business needs. Fabric also supports parallel transaction processing and a multi-channel structure, providing high performance and strong scalability [6]. Institutions such as Citibank and JPMorgan Chase have explored its application in financial transactions, payment settlements, and regulatory audits, showcasing Fabric’s strong adaptability in enterprise-level environments [3]. As shown in Figure 8:

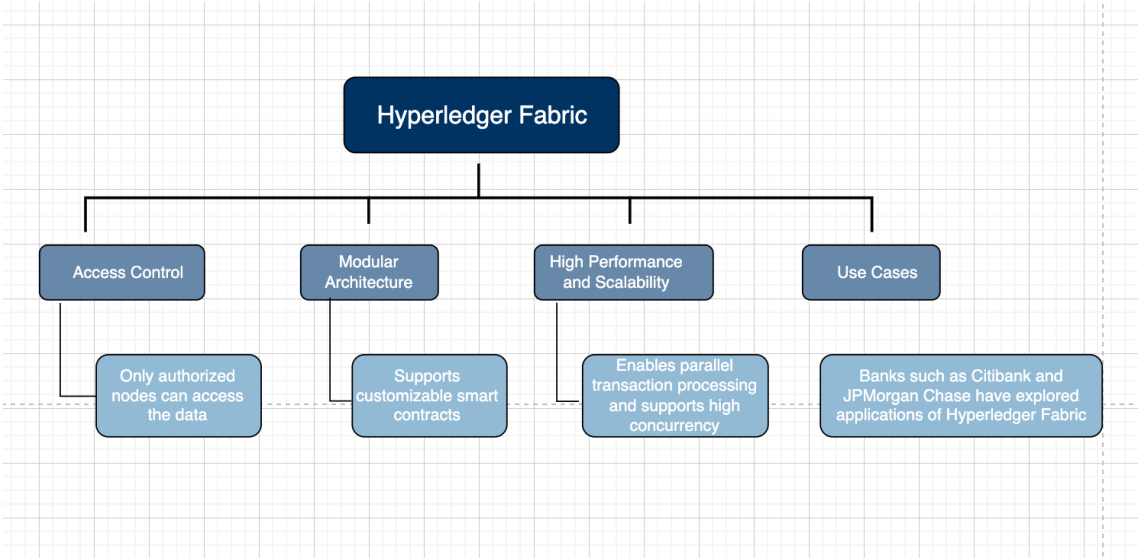

Fig.8.Hyperledger Fabric

4.2.2 Corda

Corda employs an access control mechanism that ensures transaction data is shared only among relevant parties, effectively protecting business confidentiality and user privacy [26][27]. Its support for legal contracts allows smart contracts to be integrated with traditional legal agreements, enhancing compliance and auditability. Additionally, Corda uses an efficient consensus mechanism where only the transacting parties and notary nodes validate transactions, significantly improving processing efficiency [28]. Corda has been adopted by financial institutions such as Credit Suisse and Wells Fargo for loan management and complex financial transactions, demonstrating its high applicability in the banking industry [29]. As shown in Figure 9:

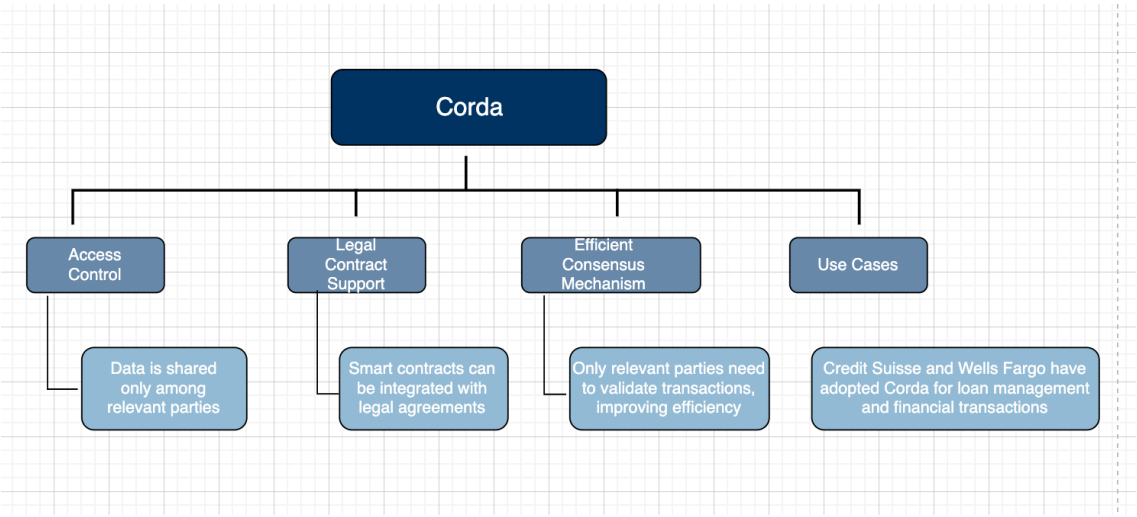

Fig.9.Corda

4.2.3 Quorum

Quorum is an enterprise-grade blockchain platform derived from Ethereum, designed specifically for financial institutions and scenarios with high privacy requirements. It enhances privacy protection by supporting private transactions and smart contracts, and data is only shared between authorized participants [30]. Quorum utilizes a separate component, Tessera, for secure transaction processing, ensuring that sensitive business information is not exposed to the entire network.

At the same time, Quorum retains Ethereum's smart contract capabilities, enabling automation of financial processes such as loan management to improve efficiency and reduce costs [30]. Leading institutions including BNP Paribas and JPMorgan Chase have adopted Quorum for use in financial transaction processing, data sharing, and regulatory compliance, demonstrating its practical value in the financial sector [31]. As shown in Figure 10:

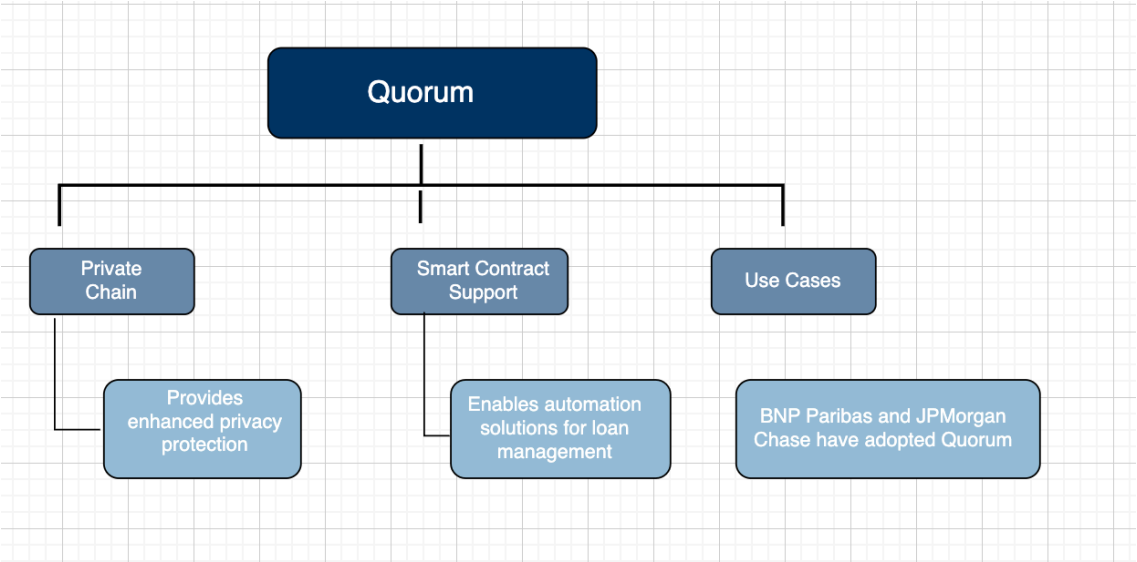

Fig.10.Quorum

Following the above analysis, Corda, Fabric, and Quorum are all enterprise-grade blockchain platforms, particularly suited for the financial industry, but they differ in privacy protection, smart contracts, and performance.

In terms of **privacy protection**, Corda shares data only among relevant transaction parties to safeguard business secrets; Fabric controls data access through authorized nodes and a multi-channel mechanism to ensure security; Quorum uses private chains to support confidential transactions, meeting high privacy requirements.

Regarding smart contracts, Corda combines smart contracts with legal agreements to enhance compliance; Fabric's modular design allows easy customization according to business needs; Quorum inherits Ethereum's smart contract capabilities, suitable for complex automated processes.

On performance, Corda improves transaction speed through an efficient consensus mechanism; Fabric supports parallel processing and multi-channel

architecture to boost scalability; Quorum optimizes Ethereum's consensus to balance security and efficiency.

In summary, Corda is suited for financial businesses focusing on privacy and compliance, Fabric is flexible and easily extensible, and Quorum fits scenarios requiring complex contracts and high privacy.

4.3 Consensus Mechanism Selection and Optimization

The consensus mechanism is the "heart" of the blockchain. It determines how nodes in the network reach agreement and validate which transactions are legitimate. Thanks to consensus mechanisms, blockchains can ensure data consistency and prevent security issues like "double spending." Moreover, different consensus mechanisms affect the system's speed, security, and level of decentralization [5][6].

When banks adopt blockchain technology, they typically prioritize fast processing, low latency, and data confidentiality. Therefore, selecting a consensus mechanism requires weighing these specific needs:

- **High-concurrency scenarios**: Consensus algorithms such as PBFT or Raft are more suitable, as they enable rapid consensus and efficiently handle a large number of transactions in trusted environments [32][33]. These algorithms are commonly used in permissioned blockchain systems like Hyperledger Fabric due to their low latency and crash fault tolerance [36].
- **Low-energy scenarios**: Proof of Stake (PoS) is considered a favorable option due to its significantly lower energy consumption compared to Proof of Work (PoW), making it better suited for sustainable and long-term financial systems [34][35].
- **High-security scenarios**: If maximum fault tolerance and decentralization are required, PoW remains a reliable choice despite its high energy costs, as it has proven secure in public blockchains like Bitcoin and Ethereum [33][34].

To meet the specific needs of the banking industry, consensus mechanisms can be optimized to become more efficient and secure. For example:

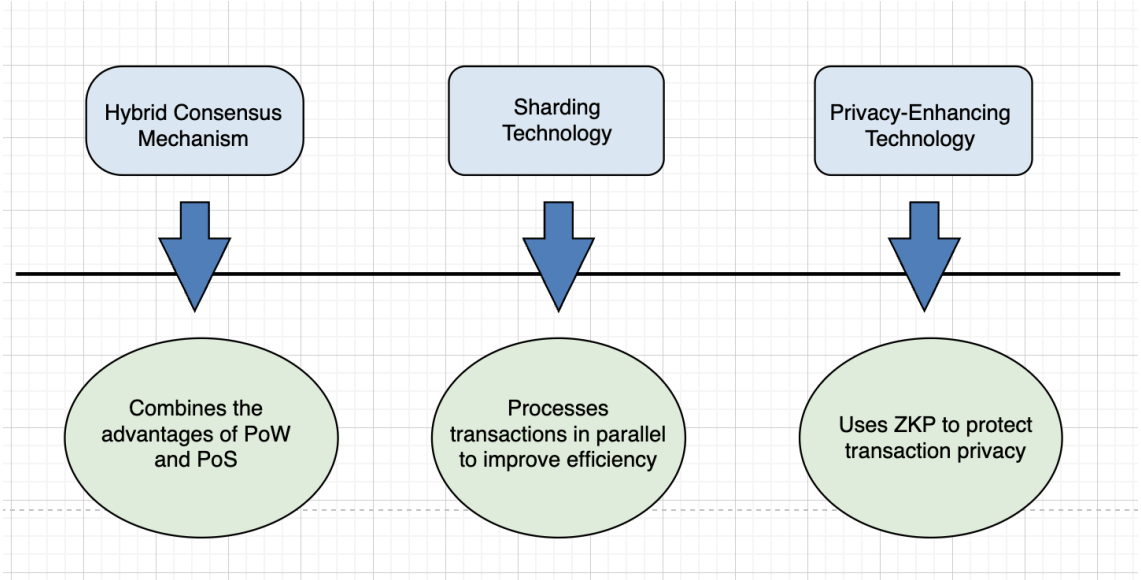

Fig.11.Consensus mechanism optimization

To overcome the limitations of a single consensus algorithm, hybrid consensus mechanisms combine Proof of Work (PoW) and Proof of Stake (PoS) — to strikesecurity [37][38][39], efficiency. For example, a system might use PoW to elect block proposers (ensuring strong Sybil resistance), while relying on PoS for fast and energy-efficient block validation. This layered approach maintains the robustness of PoW while significantly reducing its energy demands, making it especially appealing for financial applications that require both security and scalability.

Another key optimization is sharding technology, which improves performance by dividing the network into smaller units, or shards, that process transactions in parallel [40][41]. Each shard handles a portion of the total workload, greatly increasing throughput and reducing latency. In the banking sector, this allows different business divisions—such as retail, investment, and risk management—to operate on the same blockchain infrastructure simultaneously without interference, thereby enhancing scalability and operational efficiency.

4.4 Network Architecture Design and Node Management

The network architecture of a blockchain is like its foundation—it directly affects how fast the system runs, how secure it is, and whether it can grow as the business expands. For banks, a well-designed network architecture can handle lots of transactions at the same time, keep data safe, and make it easier to scale in the future. That’s why designing the network structure is a really important part of building a blockchain system. Right now, the most common types of blockchain networks are public chains, consortium chains, and private chains. We've already talked about their features in earlier sections.

When designing a blockchain network architecture, node management is a key part to consider. The **main aspects** include:

- Who can join the network (**Access Contro**l): Permissions need to be managed to decide which nodes can participate, ensuring network security and compliance.
- What each node does (**Role Assignment**): For example, some nodes store all data (full nodes), while others handle only parts (light nodes). Clear division of roles can improve efficiency.
- How to encourage node participation (**Incentive Mechanism**): Tokens or transaction fees can be used to reward nodes that actively join in consensus and help maintain the network.
- What to do when something goes wrong (**Fault Tolerance and Recovery**): There should be mechanisms to quickly detect node failures and recover operations, also to defend against malicious attacks.

In the banking sector, the key needs for blockchain are high transaction speed, strong data privacy, and smooth multi-party collaboration. So, the choice of network architecture should be based on these needs:

- For collaboration between multiple organizations, a consortium chain is ideal. It allows shared management while maintaining privacy and efficiency.Typically, consortium chains do not require incentive mechanisms because the collaboration itself provides sufficient benefit to all participants.
- For internal process optimization, a private chain works best. It's controlled by a single organization, offers high performance, and strong privacy protection.
- For use cases that require transparency (like tracking charitable donations), a public chain can be added to provide open, verifiable information.

To better support these goals, the network structure can be adjusted as follows：

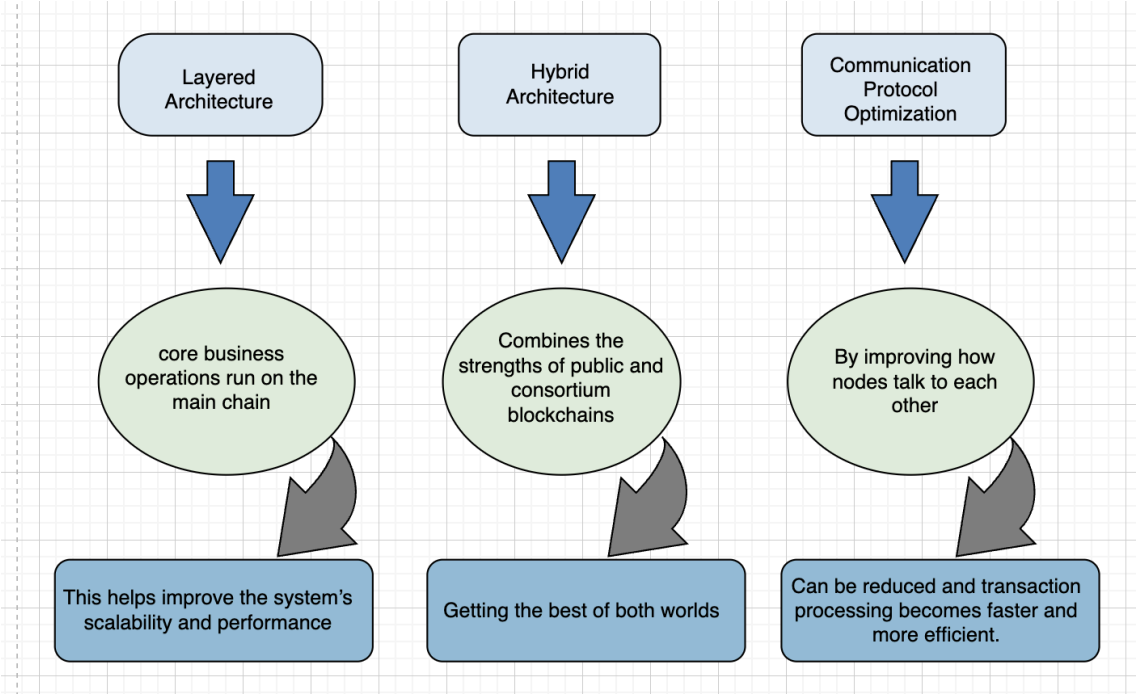

Fig.12.Network structure optimization

- **Layered Architecture**
  Layered architecture divides blockchain functionality into multiple layers, such as separating transaction records, smart contract execution, and data storage. The main chain focuses on handling core business operations, while non-core functions can be offloaded to side chains or other layers. This reduces the burden on the main chain and improves overall system performance [42][43].

- **Hybrid Architecture**
  Hybrid architecture combines the openness and transparency of public blockchains with the privacy and efficiency of consortium blockchains. For example, sensitive data can be processed privately within a consortium chain, while selected results can be published on a public chain for verification. This is ideal for banking scenarios that require both privacy and transparency [44].

- **Communication Protocol Optimization**
  Communication protocol optimization refers to improving how nodes transmit data to each other, such as using more efficient broadcasting mechanisms or better synchronization algorithms. These enhancements improve overall system responsiveness and throughput, which is especially important for handling high-concurrency transactions in banking systems [45][46].

4.5 Transaction Workflow and Data Management

The transaction process is one of the core functions of a blockchain system. It usually includes steps like initiating, verifying, packaging, and adding transactions to the chain. A well-designed process improves system

performance, ensures security, and offers a smoother user experience. So, it should be optimized based on real business needs and technical features.

Data management is also a key part of blockchain, and mainly involves:

- ✓ **Where to store data**: Important data is safer on-chain but more costly; large-scale data can be stored off-chain, but trust needs to be ensured [47][48].
- ✓ **How to protect privacy**: Encryption or privacy protocols like zero-knowledge proofs help safeguard sensitive information.For example, some cryptocurrencies operate on public chains but remain privacy-aware by using advanced cryptographic techniques such as ring signatures, stealth addresses, or zero-knowledge proofs, which hide transaction details while still ensuring network consensus [49][50][51].
- ✓ **How to keep data consistent**: Consensus mechanisms make sure all parties see the same data and avoid conflicts [47][52].
- ✓ **How to retrieve data efficiently**: Good indexing or using off-chain databases can speed up data queries [52].

For banks, the design of the transaction process needs to focus on high throughput, low latency, and data privacy. So optimizations may include:

- **High transaction volume**: Use batch processing or sharding to boost throughput [53][54].
- **Low latency needs**: Improve consensus algorithms (like PBFT) to speed up confirmation [55].
- **Privacy-focused scenario**s: Apply privacy-preserving technologies (such as zero-knowledge proofs: Zero-knowledge proofs allow one party to prove the truth of a statement to another party without revealing any additional information.) to protect sensitive data [56].

Given these requirements, the data management can be optimized in several ways:

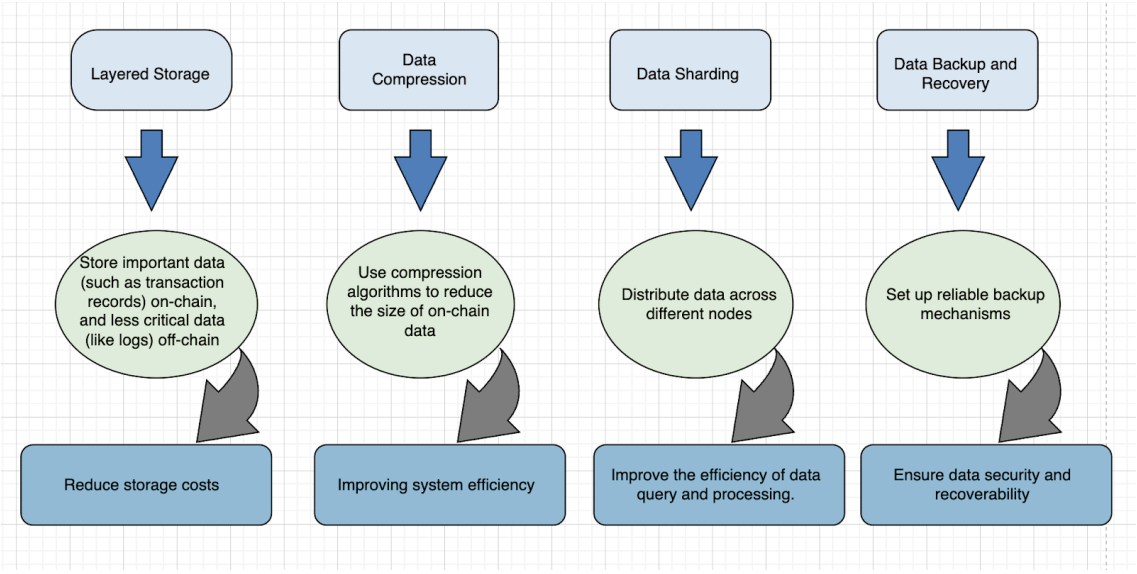

Fig.13.Data management optimization

Layered Storage

- Approach
  - Store critical data (such as transaction records) on-chain, while placing less important data (such as logs) off-chain.
- Goal
  - Reduce blockchain storage pressure and cost.
- Process Flow
  - Classify data based on importance → Keep essential data on-chain, offload non-critical data off-chain → Reduce on-chain data volume → Lower storage costs

Data Compression

- Approach
  - Apply compression algorithms to shrink the size of on-chain data, such as transaction details and smart contracts.
- Goal
  - Improve overall system performance.
- Process Flow
  - Original data → Compressed → Smaller data size → Faster processing and transmission → Increased system efficiency

Data Sharding

- Approach
  - Distribute data across multiple nodes (shards) for parallel processing.
- Goal
  - Enhance the efficiency of data querying, access, and processing.
- Process Flow
  - Split data → Assign to different shards → Process in parallel → Reduce load on individual nodes → Improve query and processing efficiency

Data Backup and Recovery

- Approach
  - Establish robust backup mechanisms, such as periodic snapshots or off-site backups.
- Goal
  - Ensure data can be recovered in the event of system failures or cyberattacks.
- Process Flow
  - Generate data → Configure backup strategies → In case of loss or failure → Recover data → Ensure data security and recoverability.

4.6 Member Onboarding and Exit Mechanism

The membership onboarding and offboarding mechanism is a key part of blockchain network management. It directly affects the system's security, compliance, and flexibility. Onboarding should consider:

- **Onboarding process**
  - The process should be clear and efficient, including steps like application, review, and approval.
  - Permission allocation: Assign different access rights based on roles. For example, regular users can only view data, while administrators can manage network settings.
- **Offboarding**
  - Exit conditions and process: Define when and how a member can leave—whether voluntarily or due to violations—and ensure that the process includes secure data cleanup.
  - Post-exit handling: Revoke access rights promptly and manage related data properly to avoid data leaks or security risks.

In addition, it is also crucial to support dynamic permission assignment tailored to each member's role and activity.

Based on these needs, we can try some smarter approaches: For example, using AI to dynamically adjust permissions can automatically optimize access based on member behavior; decentralized identity verification can further enhance privacy protection; and a modular member management design can make the system more flexible and better able to adapt to different business scenarios.

4.7 Blockchain System Maintenance and Upgrades

Routine maintenance of a distributed system involves several critical activities to ensure its stable and secure operation over time. One of the fundamental tasks is continuously monitoring the status of each node in the network. This includes checking node availability, resource usage, network connectivity, and response times. By keeping a close eye on these indicators, system administrators can quickly identify and address any anomalies or failures before they escalate into larger problems, thereby minimizing downtime and service disruptions.

In addition to monitoring, it is essential to perform regular backups of important data. These backups should be comprehensive and stored securely in multiple locations to guard against data loss due to hardware failures, software bugs, or malicious attacks. Having an effective backup and restore strategy ensures that the system can recover swiftly and resume normal operation with minimal data loss when unexpected incidents occur.

Security is another paramount concern during routine maintenance. Any discovered system vulnerabilities, whether in the underlying software, network configuration, or smart contracts, must be promptly patched and updated. Delays in applying security patches can expose the system to exploitation, leading to data breaches or unauthorized access. Therefore, maintaining a robust vulnerability management process, including timely security assessments and automated patch deployment where possible, is vital for protecting the system's integrity.

To maintain optimal performance, ongoing system optimization based on real-world usage data is also necessary. This may involve fine-tuning node parameters, optimizing communication protocols, balancing load distribution, or adjusting consensus algorithm settings. Such continuous improvements help the system adapt to changing workloads and network conditions, ensuring smooth and efficient operation even as user demand fluctuates or scales.

When it comes to system upgrades or new feature deployments, a modular approach is recommended. Breaking the system into smaller, independent components allows updates to be applied incrementally, reducing the risk of introducing bugs or causing downtime across the entire network. This method also simplifies troubleshooting by isolating issues to specific modules. Moreover, having a well-defined rollback or quick recovery plan in place ensures that if any update causes problems, the system can revert to a stable state rapidly, minimizing impact on users.

Looking ahead, as artificial intelligence (AI) technologies continue to advance, integrating AI-powered automation into operations and maintenance processes holds significant promise. Intelligent monitoring tools could analyze vast amounts of system data in real-time, predict potential failures, and even autonomously resolve common issues without human intervention [57][58]. Such automated O&M systems would enhance efficiency, reduce the need for manual oversight, and enable faster responses to incidents, thereby improving overall system reliability and user experience.

5 Proof-of-Concept (PoC) Experiment

The introduction of blockchain technology into the banking industry requires, first and foremost, validating its feasibility and effectiveness in real-world business scenarios. Therefore, the goal of this experiment is to build a simplified interbank payment network based on the Hyperledger Fabric platform. By setting up a functional blockchain environment, developing smart contracts, and simulating interbank transaction processes, we aim to assess the practical value of blockchain technology in banking operations.

5.1 Objectives and Significance of PoC Development

Main objectives include:

- Validate the feasibility of blockchain technology in interbank payment scenarios
- Construct a test network involving three banks
- Implement basic interbank payment smart contracts
- Evaluate network performance metrics (TPS, latency)

This experiment represents a prototype-level Proof of Concept (PoC). It does not aim for full-feature coverage but rather focuses on technical feasibility and preliminary system performance. The goal is to provide insights into the key challenges and opportunities of applying blockchain technology in the banking sector.

5.2 PoC Environment and Tools

5.2.1 Development Environment

Blockchain Platform

- Hyperledger Fabric 2.4.1

Development Tools

- Docker 20.10.7
- Java OpenJDK 23.0.1
- IntelliJ IDEA
- Maven 3.6.3

The entire environment is customized based on the official Hyperledger Fabric test-network example and extended according to banking business

requirements. It supports interbank payments, smart contract invocation, transaction queries, and other operations.

5.2.2 Key Components

In this experimental platform, I built a simplified version of the bank blockchain network, covering the core components required to implement a blockchain system, mainly including smart contracts, various nodes, and related configuration tools.

- **Smart Contracts**
  Smart contracts are deployed on Peer nodes, and different contract call requests will be verified and signed by multiple nodes in the network to ensure the credibility of the transaction.

- **Node compositio**
  The experimental network includes three types of nodes, each with a specific role:

    - **Peer Nodes**: Each bank (BankA and BankB) has one peer node, responsible for storing the ledger, executing smart contracts, and endorsing transactions.
    - **Orderer Node:** Acts as the “cashier” of the network, collecting all transactions, ordering them into blocks, and broadcasting them to the peers.
    - **CA (Certificate Authority) Nodes**: Handle identity management by issuing certificates to authenticate participants before they join the network.

- **Channel mechanism**
  Channels are a very practical privacy mechanism that allows different organizations to maintain their own "visible ledgers" in the same network. I created a channel named mychannel, and transactions are only conducted in this channel, and external organizations cannot see the content of these transactions.

5.3 PoC Implementation Steps

5.3.1 Building a Simple Banking Blockchain Network

- **Step 1: Start the Network and Create the Channel**

  The experimental network is initialized based on the official test-network sample provided by Hyperledger Fabric. I used the provided network.sh script with custom configurations to start the network and generate the required artifacts. As shown in Figure 14.
  This command performs several key tasks:
  1) Launches all necessary Docker containers (Orderer, Peer, CA, CLI)
  2) Generates the cryptographic materials via the Certificate Authorities (CA)
  3) Creates a default channel named mychannel
  4) Joins all peers to the channel

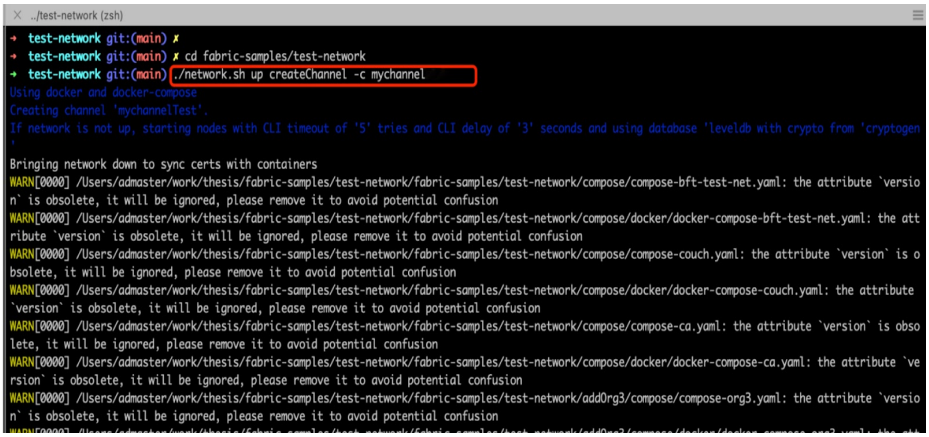

Fig.14 Start the network and create a channel

- **Step 2: Generate Cryptographic Material (MSP Certificates)**

  Membership Service Providers (MSPs) are used for identity and access control. Using the Fabric CA services, I generated certificates for each organization (e.g., BankA, BankB), including Admin and User credentials. This process ensures that only authorized entities can transact within the network. As shown in Figure 15:

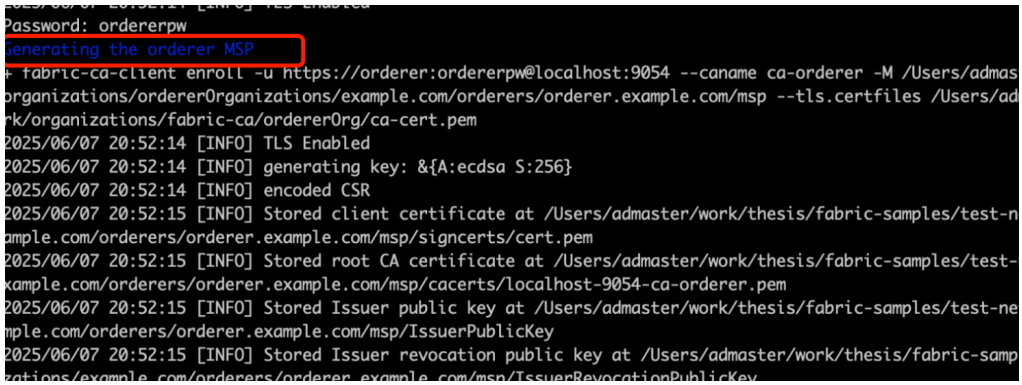

Fig.15 Generate encryption material (MSP certificate)

- **Step 3: Create a sorting service (Orderer)**

  The Orderer node is configured to manage consensus and ensure correct transaction sequencing. It uses Raft consensus in this setup. Configuration files such as orderer.yaml and configtx.yaml are prepared to define channel behavior, consortium membership, and ordering policies. As shown in Figure 16:

Fig.16 Create a sorting service (Orderer)

- **Step 4: Start the Peer node**

Each participating bank (BankA and BankB) has its own peer node. As shown in Figure 17 and these nodes are configured to:

- ➢ Host smart contracts (chaincode)
- ➢ Maintain local copies of the ledger
- ➢ Endorse transactions

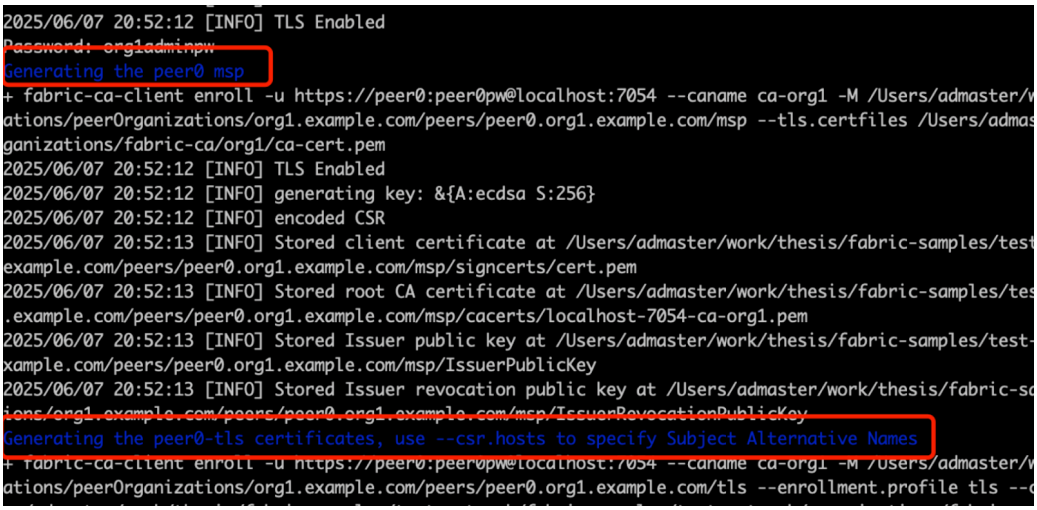

Fig.17 Start the Peer node

- **Step 5: Create a channel**

This allows all peers to participate in the same ledger state and share visibility of transactions specific to the channel. As shown in Figure 18:

Fig.18 Create a channel

- **Step 6: Deploy chain code**

The smart contract (chaincode) is written in Java and defines the logic for basic interbank transfers (e.g., transferring funds from Bank A to Bank B). As shown in Figure 19 and the deployment process includes the following:

- **Package the chaincode**: Package into the format expected by fabric.
- **Install the chaincode on all peers** : Put the chaincode package on each Peer node so that the node has the code file ready to run the chaincode.
- **Approval chain code**: Each organization agrees that the chaincode definition (version number, name, endorsement policy, etc.) meets the requirements, indicating "I approve this chaincode version".
- **Submit chaincode**: The network reaches a consensus and formally writes the chain code definition into the blockchain channel configuration, so that the entire network knows that the chain code is officially effective.
- **Initialize chaincode**

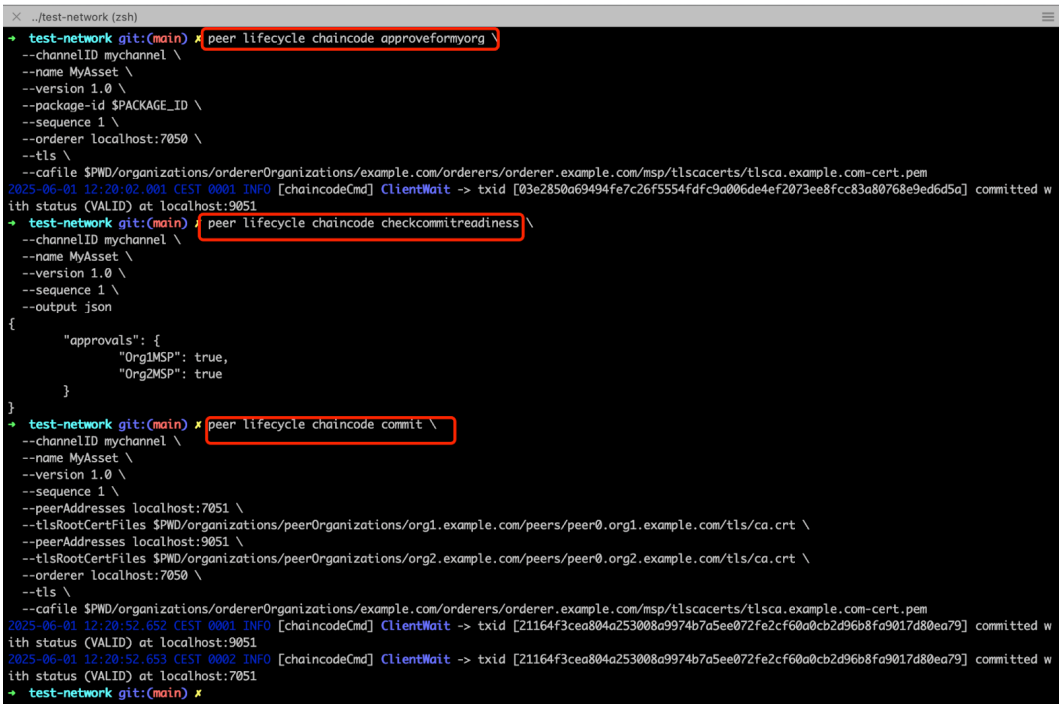

Fig.19 Deploy chain code

5.3.2 Developing Smart Contracts for Interbank Payments

In this experiment, I developed a simple bank payment smart contract (chaincode) in Java, implementing functions and as shown in fig20:

- initLedger()
  - Purpose: Initializes the ledger by creating a sample payment record
  - Implementation Details:
    - Creates a payment record with ID "p1"
    - Payer is "BankA", receiver is "BankB", amount is 1000.0

- Initial status is set to "INITIATED"
- Stores the payment record in the blockchain state database in JSON format

- createPayment()
  - Purpose: Creates a new payment record
  - Implementation Details:
    - Creates a new Payment object with the provided parameters
    - Initial status is set to "INITIATED"
    - Stores the payment record in the blockchain state database in JSON format

- approvePayment()
  - Purpose: Approves the specified payment record
  - Implementation Details:
    - Retrieves the payment record with the specified ID from the blockchain state database
    - Changes the payment status from "INITIATED" to "APPROVED"
    - Updates the payment record in the blockchain

- queryPaymentById()
  - Purpose: Queries a payment record by its ID
  - Implementation Details:
    - Directly retrieves the data of the specified ID from the blockchain state database

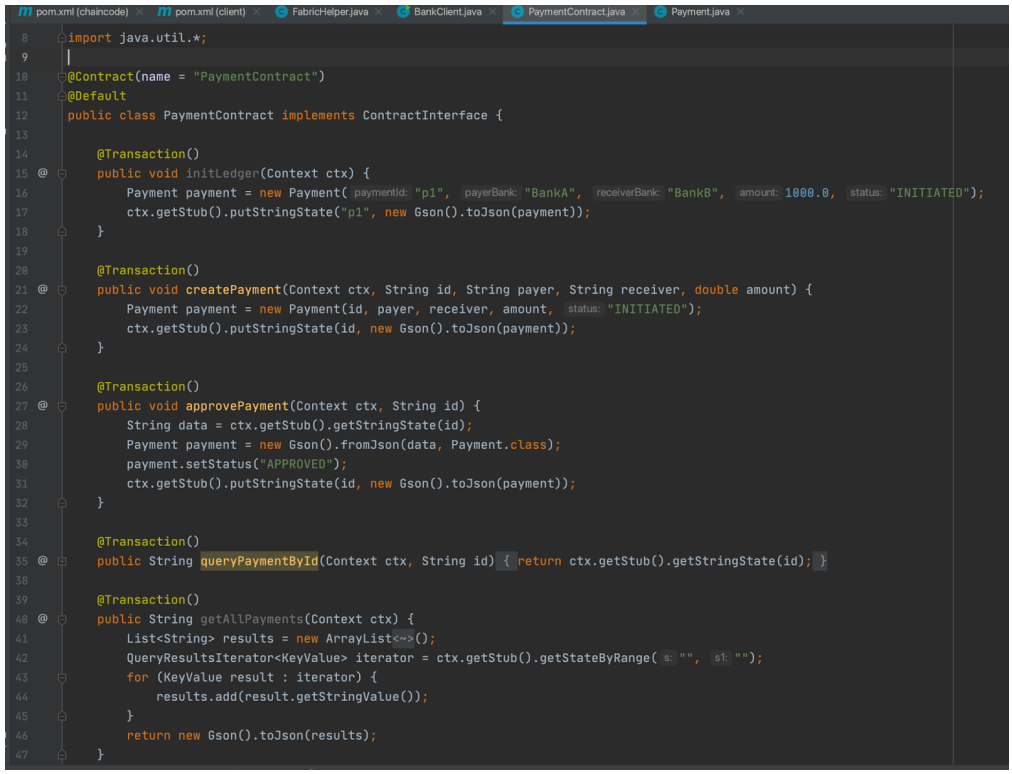

Fig.20 Method

5.3.3 Designing and Simulating Transactions for Performance Testing

500 requests were made (to transfer 1000 amount from bank A to bank B), and the time taken for each request was calculated and compared with the total time:

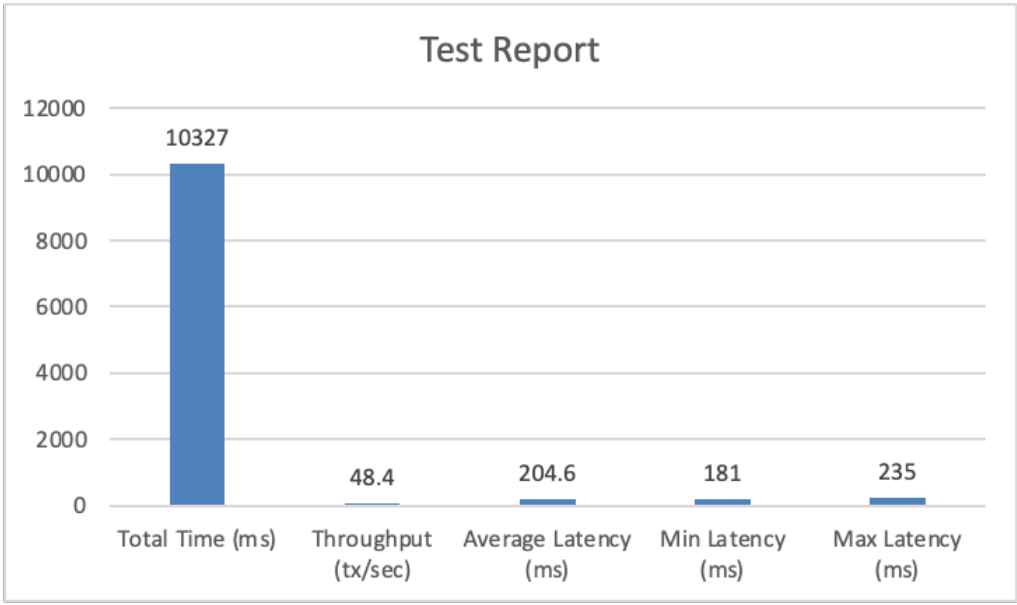

Fig.21 Test Report

5.4 PoC Experimental Results and Analysis

5.4.1 Transaction Throughput (TPS)

The observed throughput of approximately 48 transactions per second (TPS) is considered moderately high for a local Hyperledger Fabric test network environment . This performance level demonstrates that even under simplified conditions—with minimal network latency, reduced cryptographic overhead, and limited organizational complexity—the system is capable of processing transactions at a relatively efficient rate.

However, it is important to note that these results were obtained under controlled, non-production settings. In real-world deployment scenarios, additional security and validation mechanisms are typically enforced to ensure the robustness, confidentiality, and integrity of the system. These mechanisms include, but are not limited to, TLS (Transport Layer Security) encryption for secure communication, multi-organization endorsement policies to ensure consensus across different stakeholders, and orderer services with more complex consensus protocols (such as Raft or BFT) for reliable block generation and sequencing.

Each of these features introduces additional computational and network overhead. For example, TLS encryption requires extra processing for message encryption and decryption; endorsement policies require multiple peers across

organizations to validate and sign transactions, increasing cross-network communication; and advanced ordering services may introduce delays to guarantee proper block ordering and fault tolerance. As a result, the overall throughput in a production environment is generally expected to be lower than that observed in a simplified local test network.

Despite this anticipated performance reduction, such trade-offs are necessary to meet the security, compliance, and reliability requirements of enterprise-level banking applications. Therefore, when evaluating throughput for production readiness, it is essential to account for these real-world factors and conduct further stress testing under representative deployment conditions.

5.4.2 Transaction Confirmation Time

The average confirmation time recorded during the experiment was approximately 204.6 milliseconds, with a minimum latency of 181 milliseconds and a maximum latency of 235 milliseconds. This means that, on average, each transaction took about 200 milliseconds from the moment it was submitted to the network until it was successfully committed to the ledger. This relatively short end-to-end processing time reflects the efficiency of the underlying consensus mechanism and the responsiveness of the peer nodes.

Moreover, the narrow latency range — a difference of only about 54 milliseconds between the maximum and minimum values — suggests that the system maintained a high level of performance consistency and stability throughout the testing period. This consistency indicates that the chaincode execution and transaction processing were not significantly affected by system noise, transient load spikes, or network delays.

Such stable performance is particularly important for financial applications like interbank payments, where predictable confirmation times help ensure smooth operations and enhance user trust. The results also imply that under the tested conditions, the Hyperledger Fabric network can offer reliable real-time transaction processing with minimal variance, making it a viable platform for time-sensitive banking use cases.

5.4.3 Security and Privacy Protection

In terms of security, Fabric's "permissioned chain" design naturally has a strong identity control mechanism. All transaction participants need to hold legal certificates, and unauthorized users cannot enter the network.

At the same time, strong privacy protection is achieved through the following mechanisms:

- **Channel mechanism**: BankA and BankB's transactions are only visible within the bankchannel and cannot be accessed by other organizations;
- **Chaincode access control**: Through the identity verification embedded in the chaincode, it is ensured that only authorized banks can perform specific operations.

5.5 Summary of PoC Findings

This experiment only implemented the basic functionalities of Hyperledger Fabric, verifying the fundamental performance of the network and chaincode. However, real-world business scenarios are usually far more complex than the test environment, involving more business logic, access control, and data consistency requirements. Therefore, future optimizations can focus on the following aspects:

- **Optimizing smart contract execution logic**
  By streamlining and optimizing the chaincode, unnecessary computations and state reads/writes can be reduced to improve execution efficiency. At the same time, designing data models and access patterns reasonably helps lower resource consumption during chaincode execution.
- **Introducing transaction batch processing**
  By grouping multiple transactions for batch submission and endorsement, network communication and ledger write overhead can be reduced, thereby improving overall throughput and system responsiveness.

6 Conclusion and Future Work

6.1 Summary of Research Findings

This research provides a comprehensive examination of the potential applications and practical challenges of blockchain technology in the banking sector. Initially, it offers an in-depth analysis of the fundamental blockchain data structures—such as block headers, hash pointers, and Merkle trees—emphasizing their crucial role in ensuring ledger immutability, data integrity, and consistency within distributed systems. This foundational understanding clarifies why blockchain is regarded as a secure and transparent solution for financial recordkeeping in banking.

Following this, the study explores the prospects and challenges of blockchain adoption in banking from four key dimensions: first, improving operational efficiency by reducing intermediaries and lowering transaction costs; second, enabling transformation of future banking business models to foster more flexible and innovative financial services; third, supporting the development and deployment of Central Bank Digital Currencies (CBDCs), which are vital to the growth of the digital economy; and fourth, leveraging blockchain's unique capabilities in data management and privacy protection to enhance user data security and regulatory compliance.

Subsequently, a systematic comparative analysis of mainstream blockchain platforms—including public, consortium, and permissioned blockchains—was conducted. This evaluation focused on their differences in performance, security mechanisms, scalability, and suitability for specific banking use cases. Particular attention was paid to how these platforms address transaction privacy, identity verification, and regulatory compliance, thereby assisting banks in selecting the most appropriate blockchain solutions.

To validate the theoretical findings, the research team set up a localized Hyperledger Fabric test network and conducted practical experiments with simplified smart contracts. The results demonstrated that, under controlled conditions, the network could achieve stable transaction throughput (approximately 48 TPS) and low confirmation latency (around 200 milliseconds), highlighting Fabric's potential to support secure and efficient banking transactions. Although the implemented smart contracts were relatively basic, the experiments successfully confirmed the technical feasibility of using Hyperledger Fabric in interbank transaction scenarios.

Finally, the study identified several key areas requiring further optimization, including enhancing the complexity and flexibility of smart contract logic, introducing transaction batching and parallel processing mechanisms, and deepening the integration between blockchain systems and traditional banking IT infrastructure. These improvements are essential for transitioning blockchain technology from proof-of-concept experiments to production-ready deployments within the banking industry.

6.2 Key Contributions and Innovations

This study makes several key contributions to the exploration of blockchain applications in the banking sector. First, it systematically analyzes the practical value of blockchain technology in addressing longstanding challenges in banking, particularly in improving data security, transaction transparency, and operational efficiency. These enhancements are especially relevant for interbank payment systems, transaction settlements, and audit trails, where trust, traceability, and integrity are critical.

Second, the study offers a detailed technical explanation of the core data structures in blockchain systems—such as blocks, Merkle trees, and hash-linked ledgers—emphasizing how these structures contribute to data immutability, tamper resistance, and consistency in a distributed environment. This foundational analysis not only clarifies how blockchain achieves trust without central authorities but also provides valuable insights for financial engineers and IT architects seeking to design secure, transparent banking solutions.

Finally, by implementing and testing a prototype smart contract using Hyperledger Fabric, the study demonstrates how permissioned blockchain platforms can be tailored to meet specific banking requirements, such as identity control and data isolation through channel and access control mechanisms. This practical validation helps bridge the gap between conceptual blockchain advantages and their real-world feasibility in regulated financial environments.

6.3 Research Limitations

While this study successfully demonstrates the fundamental application of blockchain technology in banking using a simplified Hyperledger Fabric setup, there are several limitations that should be acknowledged.

First, the smart contract (chaincode) developed in this study is relatively basic. It lacks support for advanced access control mechanisms such as role-based access, dynamic policy evaluation, and fine-grained attribute-based permissions, which are often required in real banking environments. These are essential for deploying blockchain in regulated financial institutions.

Furthermore, exception handling is limited in scope. The current implementation does not fully account for real-world scenarios such as transaction rollbacks, network failures, dispute resolution, or cross-channel data reconciliation.

From a platform perspective, while Hyperledger Fabric offers a rich set of modular features, this study only utilized a subset of its capabilities. Advanced configurations such as private data collections, off-chain data management, TLS-based identity rotation, and pluggable consensus algorithms (e.g., Raft or BFT-based ordering services) were not explored. Additionally, performance metrics were derived from a local test network with limited nodes and traffic, which does not fully reflect the scalability and robustness challenges in a production-grade deployment.

Lastly, security mechanisms such as encryption key lifecycle management, denial-of-service resistance, and long-term data integrity assurance were not

deeply investigated. As a result, the findings of this study should be interpreted as a proof-of-concept rather than a comprehensive implementation..

6.4 Future Research Directions

This study validated the basic performance and feasibility of applying blockchain technology in the banking sector through experiments conducted on a local Hyperledger Fabric environment. The results demonstrated relatively high throughput and stable transaction confirmation times under simplified conditions. However, it is important to recognize that real-world banking scenarios are significantly more complex, involving stricter regulatory requirements, higher volumes of concurrent users, and integration with existing legacy systems.

To bridge the gap between experimental validation and practical deployment, future research should focus on constructing multi-organization, production-grade blockchain networks that replicate real banking ecosystems. This includes the implementation of robust identity and access management, dynamic endorsement policies, and scalable consensus mechanisms. Optimizing smart contract (chaincode) design for efficiency and auditability is also critical to support complex financial workflows.

Additionally, exploring transaction batching and parallel processing techniques could help improve overall system performance and reduce operational costs. Enhancing interoperability between blockchain platforms and core banking systems, such as payment gateways, risk control modules, and KYC/AML databases, will be vital for seamless adoption. Finally, long-term evaluation of network resilience, data privacy protection, and cyber-attack resistance should be conducted to ensure the blockchain infrastructure meets the security and compliance standards required in the banking industry.

7 References

[1] Y. Guo and C. Liang, “Blockchain application and outlook in the banking industry,” 2016.

[2] V. Rajnak and T. Puschmann, “The impact of blockchain on business models in banking,” 2021.

[3] L. Cocco, A. Pinna, and M. Marchesi, “Banking on blockchain: Costs savings thanks to the blockchain technology,” 2017.

[4] T. Zhang and Z. Huang, “Blockchain and central bank digital currency,” 2022.

[5] G. Tripathi, M. Ahad, and G. Casalino, “A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges,” Dec. 2023.

[6] M. Chowdhury, K. Suchana, S. Alam, and M. Khan, “Blockchain application in banking system,” 2021.

[7] Ripple, "Santander partners with Ripple to bring certainty and speed to international payments," *Ripple Insights*, Mar. 23, 2018.

[8] S. Thomas and E. Schwartz, "A protocol for interledger payments," *Cryptology ePrint Archive*, Report 2016/1007, 2016.

[9] A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," *Cryptology ePrint Archive*, Report 2016/633, 2016.

[10] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Blockchains for business process management – Challenges and opportunities," *ACM Trans. Manage. Inf. Syst.*, preprint, 2016.

[11] Ethereum Foundation, “The Merge,” Ethereum.org, 2022.

[12] Wikipedia, “Central bank digital currency,” 2025.

[13] Y. Chu 等, “Review of Offline Payment Function of CBDC Considering Security Requirements,” *Appl. Sciences*, 2022.

[14] IMF Working Paper, “CBDC and Bank Disintermediation in a Portfolio Choice Model,” 2023.

[15] ResearchGate, “Understanding the Roles and Implications of CBDCs…,” 2025.

[16] 57Blocks, “Blockchain Encryption Technology: A Critical Defense for Protecting Digital Assets,” Jan. 10, 2025.

[17] “Blockchain Security Mechanism Design Based on Chinese Cryptosystem SM2 Algorithm,” *MDPI Mathematics*, vol. 11, no. 14, 3036, 2023; and Wikipedia, “Public-key cryptography,” 2025.

[18] Investopedia, “ZK-SNARK: Definition, How It’s Used in Cryptocurrency, and History,” 2018; R. Lavin *et al.*, “A Survey on the Applications of Zero-Knowledge Proofs,” arXiv, Aug. 2024; Wikipedia, “Non-interactive zero-knowledge proof,” 2025.

[19] IMF, "FinTech Notes: Cyber Resilience of the Central Bank Digital Currency Ecosystem," 2024. (accessed Jun. 2025).

[20] X. Liu et al., "Research on the risk of block chain technology in Internet finance supported by wireless network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, Art. No. 1685, 2020.

[21] European Commission, “Legal and regulatory framework for blockchain,” Digital Strategy EC, updated 25 Oct. 2024.

[22] Gartner, via DorApp blog, “Comparison: Cloud vs. On-Premise Solutions,” Apr. 2025. (accessed Jun. 2025).

[23] MindBlender, "Cloud vs. On-Premise: Which Solution Is Right for Your Bank," 2024. (accessed Jun. 2025); 360Factors, "Cloud and On-Premise Bank Tech Security," 2024. (accessed Jun. 2025).

[24] Bitcoin101 Blog, “Blockchain vs Databases: Pros, Cons, and Real-World Applications,” 2025. (accessed Jun. 2025).

[25] “Ethereum,” Wikipedia (performance section); plus DebutInfoTech article on scalability. (accessed Jun. 2025).

[26] R3 Corda documentation, “Transaction Privacy Enhancements,” 2025. (accessed Jun. 2025).

[27] R3 Corda documentation, “Notaries,” 2025. (accessed Jun. 2025).

[28] MDPI, “A Survey of Consortium Blockchain and Its Applications,” 2024. (accessed Jun. 2025).

[29] Finextra. Credit Suisse drives blockchain project in syndicated loans market. Finextra, 28 Sept 2016.

[30] ConsenSys, “ConsenSys Acquires J.P. Morgan’s Quorum to Advance Enterprise Blockchain Adoption,” Aug. 26, 2020. [Online]. Available: https://consensys.net/blog/news/consensys-acquires-jp-morgan-quorum/. (accessed Jun. 2025).

[31] BNP Paribas, “BNP Paribas completes its first real-time Blockchain payments,” press release, 202x. (accessed Jun. 2025).

[32] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” in OSDI, vol. 99, no. 1999, pp. 173–186, 1999.

[33] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 2017, pp. 557–564.

[34] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A Survey on the Security of Blockchain Systems,” Future Generation Computer Systems, vol. 107, pp. 841–853, 2020.

[35] V. Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform,” White Paper, 2014.

[36] D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm,” in 2014 USENIX Annual Technical Conference (USENIX ATC 14), Philadelphia, PA, USA, 2014, pp. 305–319.

[37] F. Saleh, “Blockchain Without Waste: Proof-of-Stake,” Review of Financial Studies, vol. 34, no. 3, pp. 1156–1190, 2021.

[38] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol,” CRYPTO 2017, LNCS vol. 10401, pp. 357–388, 2017.

[39] Ethereum Foundation, “Casper the Friendly Finality Gadget,” 2017.

[40] M. Zamani, M. Movahedi, and M. Raykova, “RapidChain: Scaling Blockchain via Full Sharding,” ACM CCS, pp. 931–948, 2018.

[41] W. Wang et al., “A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks,” IEEE Access, vol. 7, pp. 22328–22370, 2019.

[42] W. Wang, D. T. Hoang, P. Hu, D. Niyato, P. Wang, and Y. Wen, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” IEEE Access, vol. 7, pp. 22328–22370, 2019. DOI: 10.1109/ACCESS.2019.2896108

[43] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in Proc. IEEE Int. Congress on Big Data, 2017, pp. 557–564.

[44] X. Xu, I. Weber, M. Staples, et al., “A taxonomy of blockchain-based systems for architecture design,” in Proc. IEEE Int. Conf. on Software Architecture (ICSA), Gothenburg, Sweden, 2017, pp. 243–252. DOI: 10.1109/ICSA.2017.33

[45] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains,” in Proc. ACM SIGSAC Conf. on Computer and Communications Security (CCS), Vienna, Austria, 2016, pp. 3–16. DOI: 10.1145/2976749.2978341

[46] S. Bano, A. Sonnino, M. Al-Bassam, et al., “Sok: Consensus in the age of blockchains,” ACM Computing Surveys (CSUR), vol. 52, no. 2, pp. 1–34, Apr. 2019. DOI: 10.1145/3316481

[47] W. Wang et al., “A survey on consensus mechanisms and mining strategy management in blockchain networks,” IEEE Access, vol. 7, pp. 22328–22370, 2019. DOI: 10.1109/ACCESS.2019.2896108

[48] Z. Zheng et al., “An overview of blockchain technology: Architecture, consensus, and future trends,” Proc. IEEE Int. Congress on Big Data, 2017, pp. 557–564.

[49] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” Business & Information Systems Engineering, vol. 59, no. 3, pp. 183–187, 2017. DOI: 10.1007/s12599-017-0467-3

[50] E. Ben-Sasson et al., “Zerocash: Decentralized anonymous payments from Bitcoin,” in IEEE Symposium on Security and Privacy, 2014. DOI: 10.1109/SP.2014.36

[51] J. Chen and M. Xu, “Blockchain-based data indexing and query system: A survey,” Future Generation Computer Systems, vol. 120, pp. 76–90, 2021. DOI: 10.1016/j.future.2021.01.009

[52] J. Chen and M. Xu, “Blockchain-based data indexing and query system: A survey,” Future Generation Computer Systems, vol. 120, pp. 76–90, 2021. DOI: 10.1016/j.future.2021.01.009

[53] L. Luu et al., “A secure sharding protocol for open blockchains,” in ACM SIGSAC Conf. on Computer and Communications Security, 2016.

[54] A. Kokoris-Kogias et al., “OmniLedger: A secure, scale-out, decentralized ledger via sharding,” in IEEE Symposium on Security and Privacy (SP), 2018.

[55] Y. Xiao et al., “A survey of distributed consensus protocols for blockchain networks,” IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432–1465, 2020.

[56] S. Bowe, A. Gabizon, and I. Miers, “Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model,” in USENIX Security Symposium, 2018.

[57] Z. Zheng, H.-N. Dai, and Q. Wen, “Blockchain intelligence: When blockchain meets artificial intelligence,” arXiv preprint arXiv:1912.06485, Dec. 2019.

[58] R. C. Shit and S. Subudhi, “AI-powered anomaly detection with blockchain for real-time security and reliability in autonomous vehicles,” arXiv preprint arXiv:2505.06632, May 2025.

8 Appendices

Project code: https://github.com/XueCheng5676/Thesis