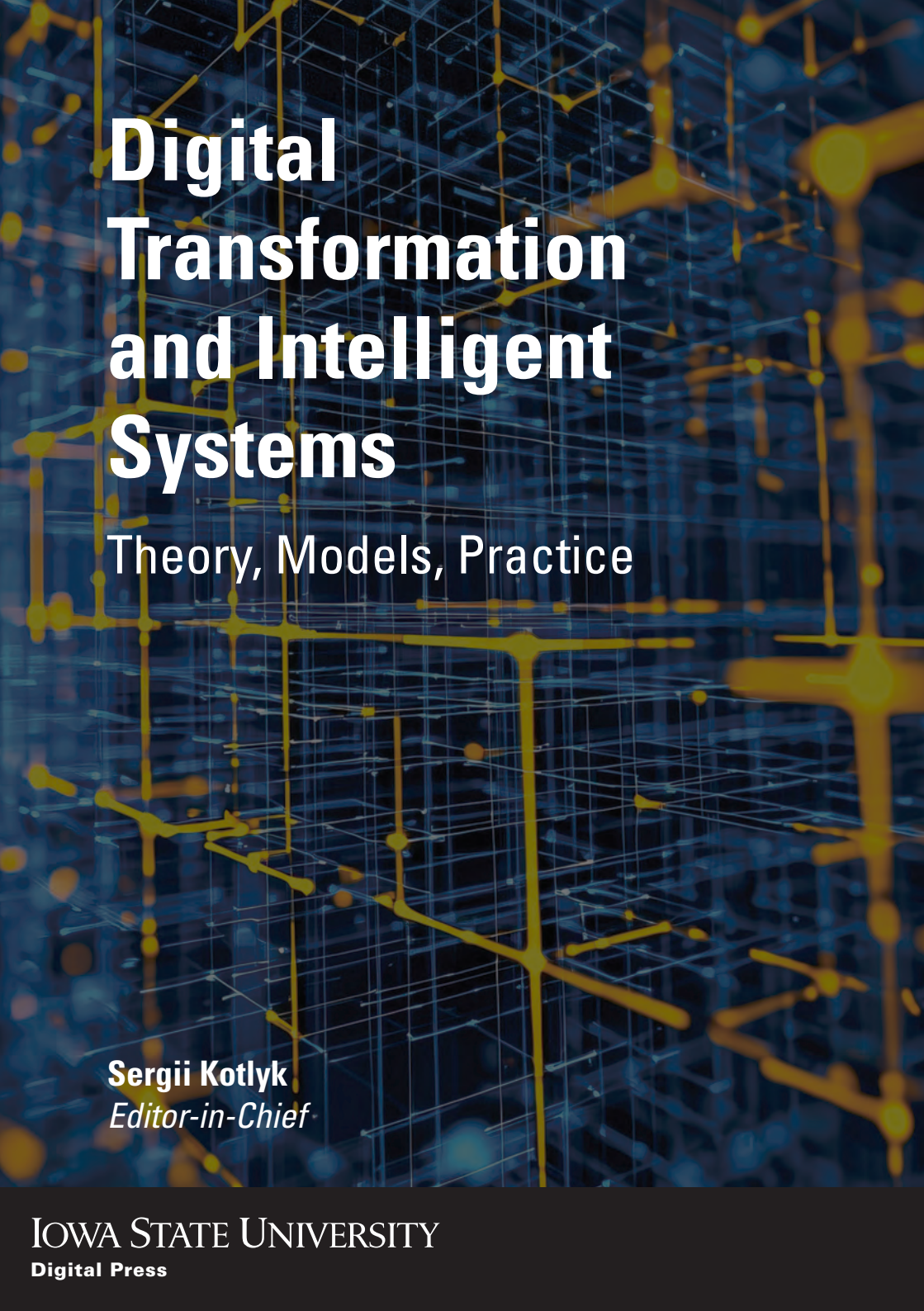




Digital Transformation and Intelligent Systems

Theory, Models, Practice

Sergii Kotlyk
Editor-in-Chief

IOWA STATE UNIVERSITY
Digital Press

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

Odesa National University of Technology

DIGITAL TRANSFORMATION AND INTELLIGENT SYSTEMS: THEORY, MODELS, PRACTICE

Monograph
Scientific publication (issue)

Editor-in-Chief
Sergii Kotlyk

Iowa State University Digital Press
2025

UDC 004.01/08
D

This monograph summarizes and analyzes the current state of scientific research in the field of information technology, mathematical modeling, automation of production processes, blockchain architectures and artificial intelligence. It considers issues of system optimization, the use of ontologies in architectural modeling, process control, the development of new visualization methods, and the use of large language models for automatic data processing.

The monograph will be useful for both specialists and employees of companies working in the IT field, as well as teachers, masters, students and post-graduates of higher educational institutions, as well as everyone interested in modern trends in the field of digital technology.

The monograph was prepared based on the results of the XVII International Scientific and Practical Conference “Information Technology and Automation — 2024”, held in October 2024 at the Odessa National Technological University.

Reviewers:

Sergey Artemenko, Doctor of Technical Sciences, Professor, Head of the Department of Computer Engineering, Odesa National University of Technology;

Yuriy Nitsuk, Doctor of Physical and Mathematical Sciences, Professor, Dean of the Faculty of Mathematics, Physics and Information Technologies of the Odessa I. I. Mechnikov University;

Sergey Pavlov, Doctor of Technical Sciences, Professor, Professor of the Department of Biomedical Engineering and Optical-Electronic Systems of the Vinnytsia National Technical University, Vice-President of the International Academy of Applied Radio Electronics Sciences

Editor-in-Chief **Sergii Kotlyk**, Ph.D., Associate Professor, Odesa National University of Technology (Odessa, Ukraine)

Team of Authors: **Brynza Natalia, Chaplinskyy Yuriy, Kaverinskiy Vladislav, Khoshaba Oleksandr, Kotlyk Sergii, Malakhov Kyrylo, Romaniuk Oksana, Romanyuk Oleksandr, Shvets Dmitriy, Sokolova Oksana, Stepanov Michael, Voinova Svitlana, Zavalniuk Yevhen**

Recommended by the Academic Council of the Odesa National University of Technology (*Protocol № 10 dated 04.03.2025*)

ISBN: 978–1–958291–11–5

© Brynza N., Chaplinskyy Y.,
Kaverinskiy V., et al, 2025

8. NOTATIONS OF ARCHITECTURAL MODELS IN BLOCKCHAIN NETWORKS

DOI: 10.31274/isudp.2025.197.08

Oleksandr Khoshaba

Abstract. This paper is devoted to substantiating the need to study architectural models of blockchain networks, which include local single-node, local multi-node, distributed single-node, and distributed multi-node structures. An analysis of their key characteristics is given, considering such parameters as scalability, security, and decentralization. These parameters constitute the fundamental concept of the blockchain trilemma, which lies in the need to compromise between them. At the same time, the paper analyzes the relationship of the blockchain trilemma with architectural models of blockchain networks, including local single-node, local multi-node, distributed single-node, and distributed multi-node structures. It is shown how each architectural model implements the specified parameters and encounters the limitations of the blockchain trilemma. Also, the main application areas and tasks solved using each model are given based on the proposed notations of architectural models of blockchain networks. This approach contributes to the development of blockchain technologies and their adaptation to various research areas.

Keywords: blockchain, architectural models, architectural model notations, scalability, security, decentralization.

8.1. INTRODUCTION

Blockchain technologies are rapidly developing and applied in various fields, from finance to logistics and education. To effectively use these technologies, it is necessary to understand the architectural features of blockchain networks, which significantly affect their performance, security, and level of decentralization. Studying architectural models is essential in optimizing existing solutions and developing new approaches.

8.2. PROBLEM STATEMENT

Blockchain networks are complex distributed systems that require thoughtful architectural design to ensure performance, security, and decentralization. Notations are introduced to classify architectural solutions and simplify the analysis and modeling process. However, their application in practice reveals many problems related to scalability, consensus implementation, interaction complexity, and resource limitations. These problems require careful research and development of approaches to eliminate them.

Let's consider the problems of designing blockchain network architectures. One of the common ones is the limited scalability of local structures (*SL/BL/DN* and *SM/BM/DL* notations). Although such architectures are often used in developing smart contracts, they describe structures with low or medium scalability, making it difficult to use them in real-world scenarios with high load. Choosing such architecture for developing distributed software applications leads to significant errors. At the same time, problems arise where increased transactions lead to decreased performance and delays in blockchain networks. Therefore, local structures that depend on the performance of a single server are an inapplicable solution for tasks with high *TPS* requirements. The solution to such problems is using more complex architectures such as *SH/BH/DH* with load distribution between nodes or developing hybrid models combining local and distributed solutions. Thus, architectural solutions for blockchain networks based on notations effectively analyze and model the subject area. However, several problems are related to scalability, the complexity of interactions, limitations of the analysis, and practical aspects. Their solution requires an integrated approach, including the development of new models, the use of simulation platforms, and the implementation of adaptive optimization methods. These efforts aim to create more efficient, reliable, and sustainable blockchain networks that meet future task requirements.

8.3. REVIEW OF RESEARCH ON ARCHITECTURAL MODELS OF BLOCKCHAIN NETWORKS

Blockchain technology has emerged as a transformative paradigm, enabling the development of decentralized and transparent systems across various domains. However, the architectural design of blockchain networks significantly influences their efficiency, scalability, and applicability. This

review critically analyzes existing architectural models of blockchain systems, highlighting their strengths, limitations, and suitability for specific use cases.

8.3.1. Public Blockchain Architectures

Public blockchains, such as Bitcoin and Ethereum, are designed to ensure decentralization and transparency by allowing open participation and maintaining a distributed ledger [1, 2]. While these architectures excel in trustless environments, they face critical scalability and energy consumption challenges. For instance, Bitcoin's *Proof-of-Work (PoW)* consensus mechanism requires substantial computational power, leading to inefficiencies and high environmental costs [3]. Efforts to mitigate these issues, such as the transition of Ethereum to *Proof-of-Stake (PoS)*, have demonstrated potential improvements in energy efficiency. Still, questions remain about the centralization risks introduced by *PoS* mechanisms [4, 5].

8.3.2. Private Blockchain Architectures

Private blockchains, employed predominantly in enterprise settings, restrict participation to authorized entities, enabling higher transaction throughput and better data privacy [6]. However, the reliance on centralized control mechanisms undermines blockchain's fundamental ethos: decentralization. For example, Hyperledger Fabric, a widely adopted private blockchain framework, provides modular architecture but struggles with scalability and interoperability when integrating with other blockchain systems [7, 8].

8.3.3. Consortium Blockchain Architectures

Consortium blockchains aim to strike a balance between the openness of public blockchains and the control of private ones by involving multiple organizations in the decision-making process [9]. While these architectures promote collaboration, they often face challenges related to governance and trust among consortium members. The Byzantine Fault Tolerance (*BFT*) consensus mechanisms used in these networks, such as those implemented in Corda, are efficient for smaller networks but become computationally expensive as the network size increases [10, 11].

8.3.4. Layered and Modular Architectures

Layered blockchain architectures have gained attention for their ability to address scalability issues by separating the data, transaction, and consensus layers [12]. For instance, the Lightning Network introduces an off-chain layer to Bitcoin, allowing microtransactions without overloading the main chain. However, such approaches often compromise security by introducing trust dependencies on the secondary layers [13]. As seen in Polkadot, modular architectures enable interoperability between heterogeneous blockchains but face significant challenges in maintaining consistent security and consensus across parachains [14, 15].

8.3.5. Hybrid Blockchain Architectures

Hybrid architectures combine features of public and private blockchains to offer a tailored solution for specific applications, such as supply chain management and healthcare [16]. While these architectures provide flexibility, they also introduce complexity in managing access controls and ensuring data consistency across different blockchain types. For example, the VeChain platform leverages a hybrid approach but faces criticism for its semi-centralized governance model, which may lead to potential vulnerabilities [17].

8.3.6. Blockchain Interoperability Architectures

Interoperability remains a pressing issue for blockchain systems, as isolated networks limit the potential for widespread adoption. Cross-chain protocols, such as Cosmos and Polkadot, address this challenge by providing frameworks for communication between blockchains [18, 19]. Despite their promise, these architectures encounter challenges related to security vulnerabilities in inter-chain bridges and the complexity of implementing consistent consensus across multiple networks [20].

Thus, architectural models play a pivotal role in defining the capabilities and limitations of blockchain systems. While public blockchains excel in decentralization, their scalability issues remain a significant barrier. Private and consortium blockchains offer higher efficiency but compromise on decentralization. Hybrid and modular architectures provide innovative solutions to specific challenges but often introduce additional complexities. Addressing these limitations requires a concerted effort to innovate and

standardize blockchain architectures, ensuring their robustness, scalability, and adaptability for diverse applications.

Next, we will consider architectural models that can be conditionally divided into local and distributed structures to develop notations for the convenience of their design based on the tasks of using blockchain networks specified by users.

8.4. METHODOLOGY FOR STUDYING BLOCKCHAIN NETWORK ARCHITECTURES

Blockchain network architectures, such as local single-node, multi-node, and distributed structures, require deep analysis to ensure their scalability, security, and decentralization. The methodology for studying architectures should include the following:

- description of mathematical models reflecting the characteristics of the network;
- use of simulation platforms to test hypotheses;
- application of specialized tools for measuring performance and analyzing network behavior.

These methods allow for a comprehensive understanding of the characteristics of blockchain networks and their behavior in actual conditions.

Simulation platforms are necessary to reproduce the behavior of a blockchain network under controlled conditions, taking into account parameters such as throughput, network latency, and consensus time.

The problems with using the methodology of blockchain network architectures include the limitations of mathematical models. Their deterministic nature may not consider the unpredictability of real networks (e.g., attacks or dynamic load changes). At the same time, additional data is needed to calibrate the models.

The problems of simulation platforms include their high complexity of setting up large decentralized networks (for example, for *SH/BH/DH* notations) and limited support for some consensus algorithms.

The limitations of combined approaches include the labor intensity required to create a theoretical model and its calibration and verification, which require significant time and computational resources.

Thus, the methodology for studying blockchain network architectures should include mathematical models, simulation platforms, and specialized tools. Each area has advantages and limitations, so their combined use is the

most effective approach. Mathematical models allow for preliminary analysis, while simulation platforms confirm the results in practice. However, existing methodologies require revision to more fully account for dynamic factors such as attacks or failures and improve the accuracy and scalability of research.

8.5. DESCRIPTION OF ARCHITECTURAL MODELS OF BLOCKCHAIN NETWORKS

In blockchain networks, architectural models determine how nodes are organized, their interactions, and their physical placement. In this regard, architectural models of blockchain networks can be divided into four types: local single-node, local multi-node, distributed single-node, and distributed multi-node structures. Each of these models has its characteristics, advantages, and limitations, which makes their study necessary for choosing the appropriate architecture depending on specific tasks. Let's consider the main architectures of such structures. Let's start with the local single-node architecture of the blockchain network, which is a special case of the local multi-node architecture. It includes a network that consists of one node running on one physical or virtual server. Such a node performs all roles, including block creation, transaction validation, and data storage.

8.5.1. Local architectural model of blockchain networks

The local single-node architecture of the blockchain network is characterized by a high degree of simplicity in initial setup and subsequent maintenance (tabl. 8.1). All processes related to software installation, node configuration, and the blockchain network launch are performed on one physical or virtual server. The absence of the need for inter-node interaction simplifies debugging and troubleshooting and reduces the requirements for the network infrastructure. In addition, software updates and data management are limited to one node, which minimizes operating costs and reduces the likelihood of errors associated with data synchronization between nodes. Thus, the local single-node architecture is ideal for testing, development, and education.

One of the key characteristics of the local single-node blockchain network architecture is its ease of setup and maintenance. The deployment process requires minimal effort and resources since the network operates on a single node. Software installation, network parameter configuration,

and data management are limited to just one server, eliminating the need for complex coordination between multiple nodes. This simplifies the testing and debugging process and reduces the likelihood of errors related to network delays or incorrect data synchronization. For system administrators and developers, this architecture provides a convenient environment for experiments and pilot projects, as it reduces the number of variables that require control. Network support and updates are also simplified, as they are carried out centrally, without the need to synchronize or reconfigure multiple nodes. Thus, this architecture is optimal for tasks where minimizing the time and cost of deployment and maintenance are key factors.

Table 8.1

Comparative characteristics of local architectural models of blockchain networks

Parameter	Single-node	Multi-node
Ease of setup	High	Medium
Scalability	Low	Medium
Reliability	Low	Medium
Best use	Testing	Simulation

Despite all the advantages in terms of ease of setup and management, the local single-node architecture of the blockchain network has a significant limitation, namely, the inability to scale to handle large volumes of data and an increasing number of users. Since a single node performs all operations, its computing resources are the main factor determining the network performance. When the load on the system increases, for example, when the number of transactions increases, this node may experience significant difficulties in processing information in real-time. As a result, the speed of block formation and transaction confirmation decreases, significantly limiting the network's capabilities in conditions of intensive operation. In addition, the lack of data distribution between nodes increases vulnerability to failures since the failure of a single node leads to a complete loss of access to the network. These limitations make the local single-node architecture ineffective for large production systems requiring high throughput, fault tolerance, and scalability. The local single-node architecture of the blockchain network is an optimal solution for tasks related to testing, software development, and solving minor applied problems. This is due to several factors (Fig. 8.1) that ensure its convenience and effectiveness in some scenarios.

The simplified deployment process is that due to the lack of need for complex network infrastructure and inter-node communication, the local

single-node architecture is easily deployed on a single physical or virtual server. This allows developers to create test environments quickly and without significant time and material costs. In addition, such a structure reduces the number of variables to be controlled, reducing the likelihood of network configuration errors.

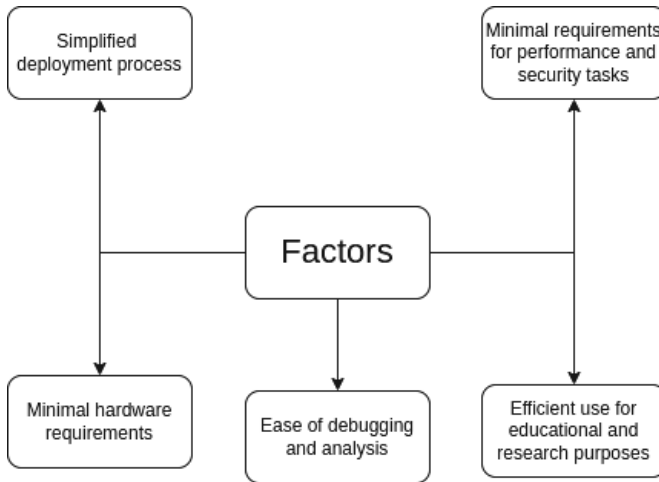


Figure 8.1. Factors that ensure the effective use of local single-node blockchain network architectures

Minimal hardware requirements are that the single-node architecture requires only one computing device, which makes it available for testing operations even on bare hardware. This allows developers and researchers to conduct experiments in resource-constrained conditions without resorting to expensive infrastructure.

The convenience for debugging and analysis is as follows. Since all processes — block generation to transaction validation — are performed on a single node, the local architecture simplifies debugging. Developers can monitor the execution of all operations in real-time and quickly identify and fix errors in software or smart contracts. This makes it an indispensable tool during the testing stages of new features and prototypes.

The local single-node architecture has compelling applications for educational and research purposes. Due to its simplicity and accessibility, it is widely used in educational and research projects. It allows students and researchers to study the principles of blockchain technology in an isolated environment

that does not require a complex setup, which contributes to effective learning and studying the theoretical aspects of blockchain technology.

The minimum requirements for performance and security tasks are as follows. The single-node architecture is sufficient for small tasks, such as data management in a local environment or implementing systems with low performance and security requirements. It can also be used to store and process a small amount of data in isolated systems or to ensure the operation of applications with a minimum number of users.

Thus, the blockchain network’s local single-node architecture is a simple, cost-effective, and effective tool for developing, testing, and solving limited-scale problems. It minimizes infrastructure costs while providing sufficient functionality to achieve the stated goals.

The local single-node architecture of the blockchain network is widely used in local innovative contract testing tasks (Fig. 8.2) and educational projects (Fig. 8.3) due to its simplicity, accessibility, and functionality. This architecture allows us to effectively explore and study key aspects of blockchain technology in a controlled environment, minimizing infrastructure and resource costs.

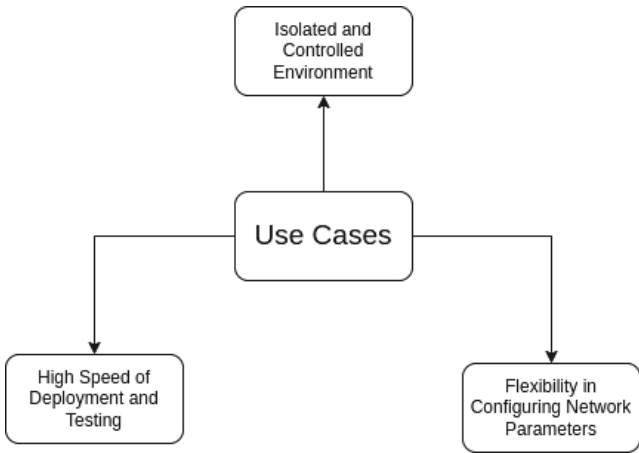


Figure 8.2. Use Cases of Local Single-Node Blockchain Network Architecture for Testing Smart Contracts

The local single-node architecture provides developers with an isolated environment for safe testing of smart contracts (Fig. 8.2). In such an environment, transactions can be simulated, the correctness of the contract

logic can be checked, and possible vulnerabilities can be identified without the risk of affecting real networks and users. This is especially important for preventing errors that can lead to financial losses or security threats.

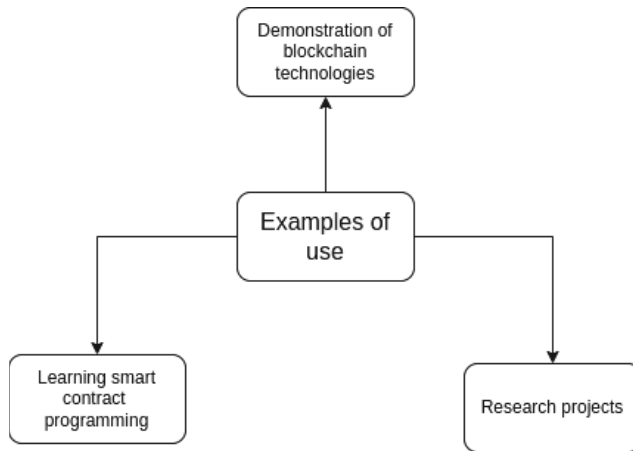


Figure 8.3. Examples of using a local single-node blockchain network architecture for educational purposes

Since all operations are performed on a single node, intelligent contract testing is much faster than distributed networks. The absence of the need to synchronize data between nodes and low transaction latency allows for prompt detection and correction of code errors, speeding up the development cycle.

Also, the local architecture can flexibly and efficiently configure blockchain system parameters, such as block generation time, mining difficulty, or consensus algorithm. This simplifies experimentation and testing of various scenarios, providing a more complete understanding of the behavior of smart contracts under different conditions.

The local single-node architecture provides the following capabilities for educational purposes (Fig. 8.3). In educational institutions, it demonstrates the basic principles of blockchain networks, including block generation, transaction validation, and network interaction. This allows students and researchers to gain practical skills in working with blockchain technologies in a safe and controlled environment.

The local network provides a convenient platform for teaching researchers how to write and deploy smart contracts. Researchers can develop sim-

ple contracts, test their execution, and analyze errors that arise. This approach contributes to a deep understanding of the basics of programming in specialized languages, such as Solidity.

The single-node architecture allows experiments to study new consensus algorithms, network interaction models, and smart contract code optimization for research purposes. Using a local environment eliminates the influence of external factors, increasing the results' reliability.

Thus, the local single-node architecture of the blockchain network is an effective tool for both developing and testing smart contracts, as well as for education and scientific research. Its use allows you to safely and quickly implement complex tasks while deepening your knowledge of blockchain technologies. Let us continue our consideration of the local architecture of the blockchain network, but let us move on to the multi-node structure. This includes a network of several nodes on one server and operating as independent network participants (Table 8.1). Such nodes interact via local IP addresses. One of the key advantages of the local multi-node architecture of the blockchain network (Fig. 8.4) regarding the distributed structure is the high performance of the transport subsystem due to the minimal delay in data transmission between nodes. Since all nodes are located on one physical server and interact via local IP addresses, the data exchange time is significantly reduced compared to distributed systems where nodes are connected via global networks. This contributes to the high block generation and validation speed, which is especially important when performing operations requiring intensive interaction between nodes, such as consensus algorithms or processing many transactions.

Minimal local network latency eliminates bottlenecks in distributed systems caused by network delays, limited bandwidth, or packet loss. This makes the local multi-node architecture optimal for simulating high-performance blockchain networks, testing consensus algorithms, and debugging network interactions under near-perfect conditions. The high performance of the transport subsystem also ensures efficient use of computing resources, reducing the likelihood of bottlenecks in data processing.

The local multi-node architecture of the blockchain network significantly simplifies the process of deploying and managing nodes through the use of containerization technologies such as Docker (Fig. 8.4). Containerization allows you to isolate each node in a separate software environment, which eliminates conflicts between dependencies and software components. This approach ensures quick setup and launch of multiple nodes on a single server without complex hardware or software configuration.

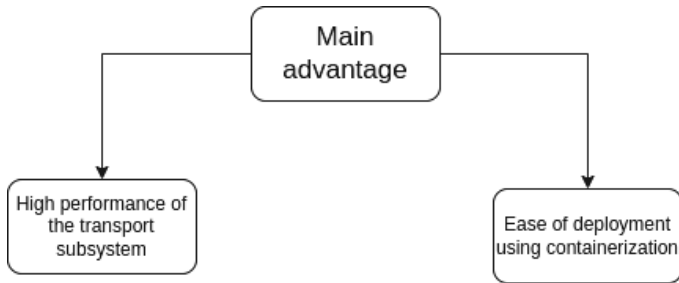


Figure 8.4. The main advantage of using a local multi-node blockchain network architecture

Docker provides ready-made images of blockchain nodes, minimizing the time spent on their installation and configuration. Scripts and container management tools can effectively automate the processes of deployment, updating, and scaling the network, which is especially important for development and testing. In addition, containerization provides high flexibility and portability, allowing you to transfer a locally deployed network to other servers or integrate it into more complex infrastructures without significant changes.

Thus, the high performance of the transport subsystem and the ease of containerization deployment make the local multi-node blockchain network architecture a powerful tool for development, testing, and research. It allows you to effectively simulate the operation of distributed systems, minimizing the cost of time and resources.

The local multi-node blockchain network architecture is an effective tool for emulating a real blockchain network, which makes it indispensable in research and development tasks related to testing consensus algorithms (Fig. 8.5) and modeling distributed systems (Fig. 8.6). By placing all nodes on a single physical server and interacting with them through local IP addresses, this architecture provides an optimal combination of ease of setup, manageability, and reliability of the simulated processes.

The local multi-node architecture allows for efficient experimentation with various consensus algorithms (Fig. 8.5), such as *Proof of Work (PoW)*, *Proof of Stake (PoS)*, or *Practical Byzantine Fault Tolerance (PBFT)*. This allows researchers to evaluate their performance, attack resistance, and behavior under high load. The local environment provides accurate and reproducible results due to minimal network latency and the absence of external influences.

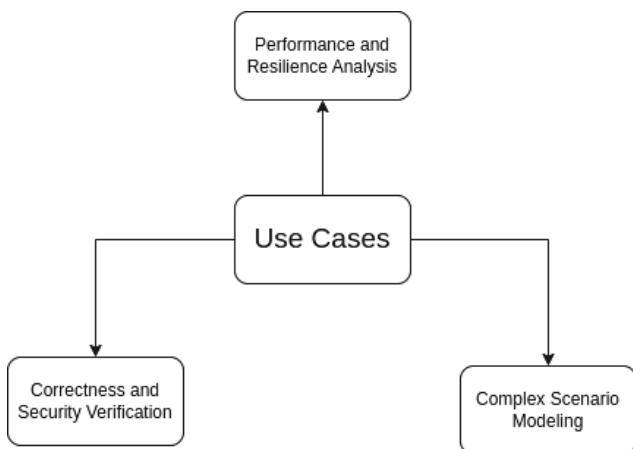


Figure 8.5. Use Cases of Local Multi-Node Blockchain Network Architecture for Testing Consensus Algorithms

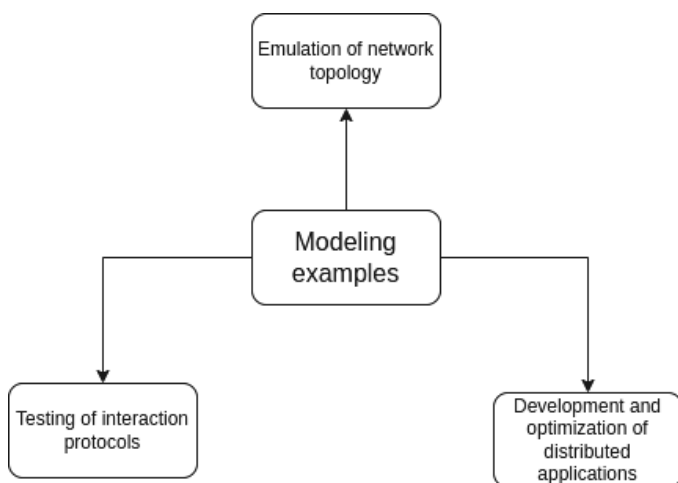


Figure 8.6. Examples of modeling distributed systems using a local multi-node blockchain network architecture

New or modified consensus algorithms can be safely tested using the blockchain network's local multi-node architecture. This allows checking their correctness and resistance to potential threats, such as Sybil or timing

attacks. The local architecture also allows for prompt changes to the node configuration and analysis of their impact on the network.

Using the local multi-node architecture of the blockchain network, it is possible to simulate various scenarios of the blockchain network, including partial shutdown of nodes, variations in data transmission delays, and distribution of computing resources. This allows us to study the behavior of consensus algorithms in conditions close to real ones without the need to deploy a large-scale distributed infrastructure.

The local multi-node architecture allows us to reproduce various topologies of distributed systems, such as star, ring, or hierarchical structures (Fig. 8.6). This is important for studying the impact of network configuration on performance and fault tolerance.

Emulation of a real network allows testing of data transfer protocols, synchronization mechanisms, and consistency between nodes. The local architecture provides a controlled environment for analyzing protocol efficiency under varying loads and network failures.

Development and optimization of distributed applications allow developers to use a local multi-node network to test distributed applications (dApps) and check their interaction with the blockchain infrastructure. This allows developers to identify performance bottlenecks and optimize the program code to improve the efficiency of distributed systems.

Thus, the advantages of using a local architecture for testing and modeling tasks are:

- a controlled environment that ensures the accuracy and reproducibility of results;
- configuration flexibility that allows you to easily change node and network parameters to conduct a variety of experiments;
- cost-effectiveness eliminates the need for a large-scale infrastructure, reducing equipment and maintenance costs.

Therefore, the blockchain network's local multi-node architecture provides powerful tools for emulating a real blockchain network, testing consensus algorithms, and modeling distributed systems. It allows you to explore complex aspects of blockchain technology in a controlled environment, minimizing costs and ensuring the results' high reliability.

8.5.2. Distributed architectural model of blockchain networks

The distributed single-node architectural model (tabl. 8.2) is a network structure in which several servers with one node each are connected via a local, corporate, or global computer network. A private, public, or heterogeneous network can be organized based on the computer network transport configuration. For these purposes, nodes in a distributed single-node architectural model interact with each other via public or private IP addresses.

Table 8.2

Comparative characteristics of distributed architectural models of blockchain networks

Parameter	Single-node	Multi-node
Ease of setup	Medium	Low
Scalability	High	Very high
Reliability	High	Very high
Best use	Early developments	Industrial networks

The distributed single-node architectural model of the blockchain network has high reliability (Fig. 8.7) due to the distributed data storage between several nodes, each of which is located on a separate server. In this architecture, each copy of the blockchain is stored on an independent node, which ensures the system's resistance to failures of individual components. Even if one or more nodes fail, the network maintains operability and access to data due to the replication of information on other nodes. This property makes the distributed single-node architecture especially valuable for implementing systems where data security and fault tolerance are critical.

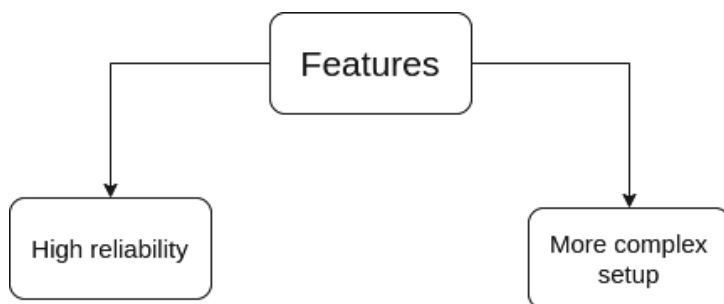


Figure 8.7. Features of the distributed single-node architecture of the blockchain network

Distributed data storage reduces the risk of data loss due to hardware failures, software errors, or external attacks. This model also protects against targeted attacks on one of the nodes since the data remains available on other network participants. Thus, a distributed single-node architecture helps to increase the system's security and resilience to external and internal threats, making it suitable for corporate and public networks with a high level of responsibility for data security.

A distributed single-node architectural model requires a more complex setup and management (Fig. 8.7) since it requires network and node connection management. Unlike local models, where nodes interact within a single server, a distributed system requires the configuration of network connections between nodes, which involves setting up public or private IP addresses, routing, and network security management. To ensure stable interaction between nodes, it is necessary to consider factors such as data transfer delays, network bandwidth, and possible communication failures. Setting up the network also includes ensuring data synchronization between nodes and the correct operation of consensus algorithms. This requires taking into account the features of the network infrastructure, such as the presence of firewalls, the use of virtual private networks (VPNs), and the configuration of encryption protocols to protect transmitted data. In addition, administrators need to monitor the status of nodes, manage their connections, and respond to potential problems, which requires additional technical expertise and resources.

The complexity of setting up and managing a distributed single-node architecture is offset by its high reliability and fault tolerance, which makes it especially relevant for large-scale blockchain systems designed to ensure security and sustainability in distributed environments.

The distributed single-node architecture of the blockchain network is an effective tool for researching and developing decentralized applications (dApps) at the initial stages of their creation. In such an architecture, each server acts as an independent node with unique tasks and functions, which allows you to create an environment as close as possible to the operating conditions of a real blockchain network. This is especially important for studying the interaction processes between nodes (Fig. 8.8), ensuring data consistency, and optimizing algorithms (Fig. 8.9). In a distributed single-node architecture. Each server can be configured to perform a specific role in the network, such as a validator, consensus participant, archive node, or relay node. This separation of functions allows us to study the impact of each role on the overall performance of the network, as well as its productivity and stability. For example, we can analyze how the load on the validator

node affects the speed of transaction confirmation or how relay nodes improve the distribution of blocks across the network.

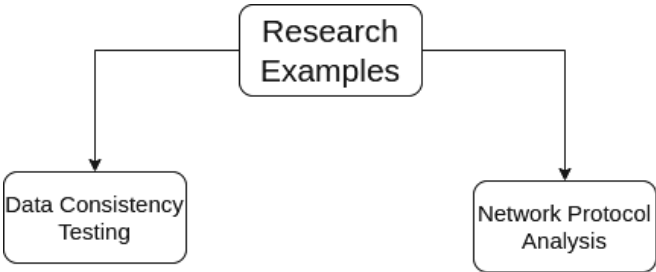


Figure 8.8. Research Examples of Node Interaction and Synchronization in a Distributed Single-Node Blockchain Network Architecture

The distributed architecture enables the simulation of key aspects of decentralization, including the distribution of computational tasks, access rights management, and transaction processing. This allows developers to practically test the interaction patterns between nodes, which is necessary to ensure the security and resilience of decentralized applications.

The distributed single-node architecture allows us to study (Fig. 8.8) how nodes synchronize data during the initial app development phase, especially under high loads or network failures. This is important for ensuring the integrity of the blockchain, as any inconsistencies can lead to forks or application failures.

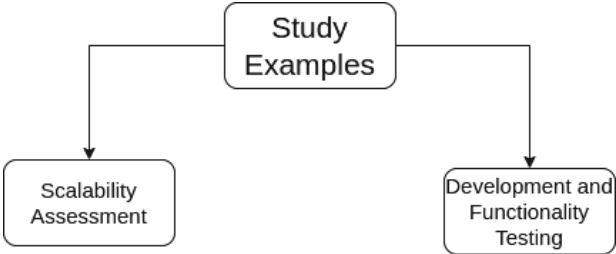


Figure 8.9. Study Examples of Blockchain Network Performance Optimization

This architecture effectively tests various network protocols, such as block and transaction protocols and encryption and routing mechanisms. It allows you to explore optimal ways for nodes to interact and identify poten-

tial network bottlenecks. At the initial stages of creating apps, it is essential to evaluate how the architecture and algorithms will scale with an increase in the number of nodes or the volume of data processed (Fig. 8.9). The distributed single-node model provides a safe and controlled environment for conducting such experiments.

dApps' functionality, including smart contract logic, interaction with external systems, and user data management, can be tested step-by-step during development. The distributed architecture allows for early error detection and debugging, reducing risks when moving to larger and more complex configurations.

Thus, the distributed single-node architecture of the blockchain network serves as a powerful tool for exploring the initial phase of creating decentralized applications. It provides a flexible and reliable environment for testing various aspects of the blockchain network, such as node interaction, data synchronization, and implementing unique functional roles. This architecture allows developers to effectively work out key elements of dApps, which helps improve the quality and reliability of the final product.

A distributed multi-node model of a blockchain network consists of nodes distributed across several servers, each containing several nodes. The nodes interact through a private, public, or heterogeneous network.

Therefore, one of the key characteristics of the distributed multi-node architecture of the blockchain network is its high resistance to failures due to the geographical distribution of nodes (Fig. 8.10). Placing nodes on different servers located in different geographical locations minimizes the risk of data loss or disruption of the network in the event of local technical failures, natural disasters or targeted attacks on the infrastructure. Even if one or more servers become unavailable, the remaining nodes will continue to provide network operation and access to data, which allows maintaining its integrity and functionality.

The geographical distribution also helps to increase the resilience of individual blockchain network services to attacks such as distributed denial of service (DDoS) attacks. Since the nodes are distributed across different regions, the network load can be redistributed, preventing overload and ensuring stable operation. Thus, this architecture guarantees high fault tolerance and reliability, making it suitable for mission-critical applications requiring high availability.

Distributed multi-node architecture plays a vital role in ensuring the absolute decentralization of the blockchain network (Fig. 8.10). Unlike architectures with a limited number of nodes located in one place, this mod-

el allows for the implementation of an even distribution of computing and network resources between network participants. Each server containing several nodes acts as an independent participant, contributing to consensus, transaction validation, and data storage processes. This eliminates the possibility of concentrating control over the network in the hands of one organization or group of people, ensuring equal rights and access to resources for all participants. True decentralization increases trust in the blockchain network, eliminating the risk of data manipulation or centralization of control. Such a structure creates stable and secure ecosystems where participants can interact without trusted intermediaries. In addition, decentralization promotes a fair distribution of rewards for participation in the network, stimulating its development and attracting new participants.

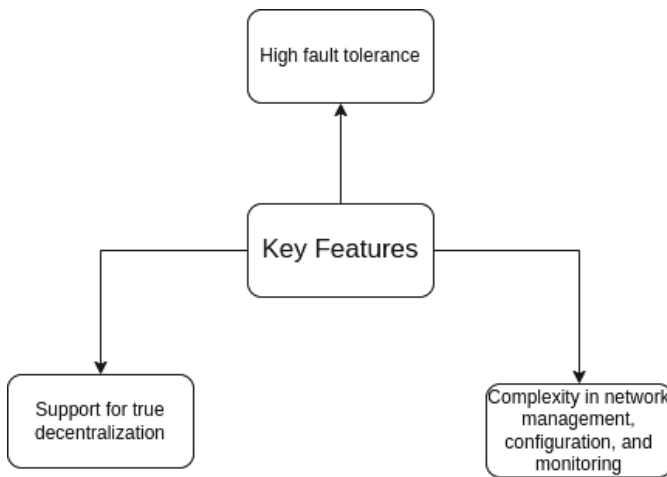


Figure 8.10. Key Features of a Distributed Multi-Node Blockchain Network Model

The complexity of managing a distributed multi-node architecture is its inherent characteristic (Fig. 8.10), as many nodes must be coordinated on different servers and in different geographic locations. Such a network configuration requires a deep understanding of network protocols, data routing, and security. It is necessary to ensure the correct configuration of private and public IP addresses, encryption for data transmission, and protection from potential attacks.

In addition, a distributed multi-node architecture requires constant monitoring of nodes' status, synchronization, and performance. Adminis-

trators must promptly identify and eliminate failures, track delays in data transmission, and ensure the stable operation of the consensus mechanism. For these purposes, specialized management and monitoring tools and software are often used, which increases operational costs and requires highly qualified technical personnel. Thus, the distributed multi-node architecture of the blockchain network combines high fault tolerance, support for real decentralization, and complexity in management. Its advantages make it a preferred choice for implementing scalable and reliable blockchain systems despite the significant technical and resource costs of their setup and maintenance.

The distributed multi-node architecture of blockchain networks is widely used in public blockchain networks such as Ethereum and Solana due to its ability to provide decentralization, scalability, and security. These networks serve as platforms for smart contracts, decentralized applications (dApps), and financial transactions, making their distributed architecture model the foundation for various scenarios.

Blockchain networks with a distributed multi-node architecture (Fig. 8.11) guarantee high stability by evenly distributing computational tasks among many nodes. This allows for the efficient processing of decentralized application requests and the execution of smart contracts without the risk of bottlenecks. For example, in the Ethereum network, nodes process transactions and maintain the *Ethereum Virtual Machine (EVM)*, which executes smart contracts, ensuring their correctness and availability.

Networks such as Solana use a distributed architecture to achieve high throughput and low latency in transaction processing. With its geographically distributed nodes and optimized consensus protocols, such as *Solana's Proof of History (PoH)*, the network can process tens of thousands of transactions per second, making it suitable for large-scale decentralized finance (*DeFi*) platforms and *NFT* marketplaces.

The distributed multi-node architecture provides a high degree of decentralization, making public blockchain networks resilient to attacks, including 51 % of attacks. The distribution of data and computing resources across many nodes prevents the centralization of power and data manipulation. This is critical for networks that serve as the foundation of global ecosystems, where trust among participants is key.

Public blockchain networks like Ethereum and Solana distribute nodes worldwide, providing 24/7 availability and high fault tolerance. Even if nodes in a particular region go down, the network continues to function thanks to nodes in other locations. This helps maintain the stability and

continuity of platforms used by millions of users. Some examples of the application of various scenarios for the distributed multi-node model of the blockchain system (Fig. 8.11) include decentralized financial systems. Let's consider them in more detail.

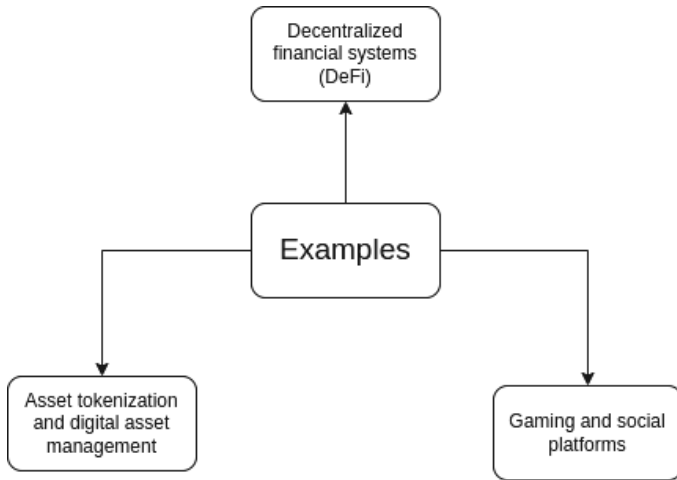


Figure 8.11. Examples of application of various scenarios of using the distributed multi-node model of the blockchain system

The distributed multi-node architecture supports *DeFi* platforms such as Uniswap and Aave, which run on Ethereum. These platforms provide users with access to lending, asset exchange, and other financial services without the involvement of traditional intermediaries.

Asset tokenization and digital asset management are represented by networks based on a distributed multi-node architecture that supports the issuance and management of tokens, including cryptocurrencies and *NFTs*. This opens up opportunities for tokenizing physical and digital assets, simplifying their transfer and management.

Gaming and social platforms that use the developments of Solana and Ethereum are also actively used to develop decentralized gaming platforms and social networks, where users can interact without centralized control, maintaining their privacy and ownership of digital assets.

Thus, the distributed multi-node architecture of the blockchain network is the basis of public blockchain networks such as Ethereum and Solana. It provides high stability, decentralization, and scalability, making it an ideal

solution for implementing various financial, social, and gaming scenarios. This architectural model continues to contribute to the development of global decentralized ecosystems, providing users with reliable and secure platforms for interaction. Using containerization technologies such as Docker in blockchain networks is a modern approach to node management that allows for efficient network scaling (Fig. 8.12) and process isolation (Fig. 8.13) in such models. Each node is launched in a separate container, which creates an isolated environment for its execution, independent of the configuration of the central system or other nodes. Using containers simplifies deploying new nodes in the network (Fig. 8.12). Pre-configured container images contain all the necessary components for the operation of a blockchain node, including software, libraries, and dependencies. This allows new nodes to be deployed in minutes, which is especially important for increasing network throughput or performing load tests.

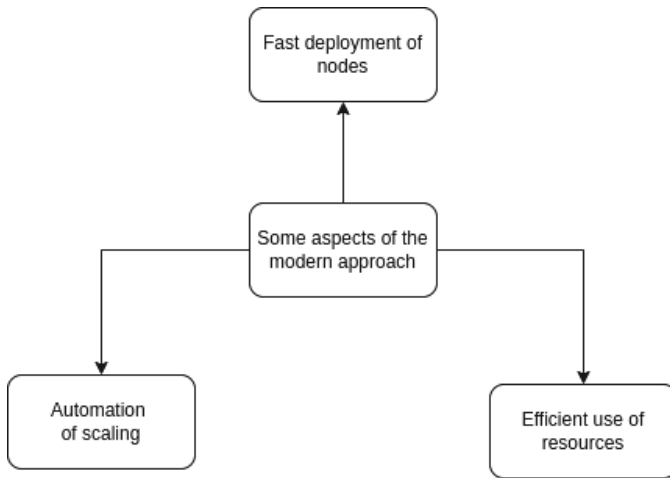


Figure 8.12. Some aspects of the modern approach to network scaling in blockchain

Containerization integrates with orchestration systems such as Kubernetes or Docker Swarm, which automatically manage network scaling (Fig. 8.12). Orchestrators add or remove nodes depending on the current network load, allowing you to dynamically adjust its performance and minimize downtime.

Containers require fewer resources than virtual machines because they use the host operating system's kernel. This allows you to run more nodes

on a single physical server while ensuring optimal performance and reducing infrastructure costs.

Each node in the container model operates in an isolated environment, which prevents conflicts between processes and dependencies (Fig. 8.13). This isolation minimizes the risks associated with data leaks or failures in one node's operation, as they do not affect other network components.

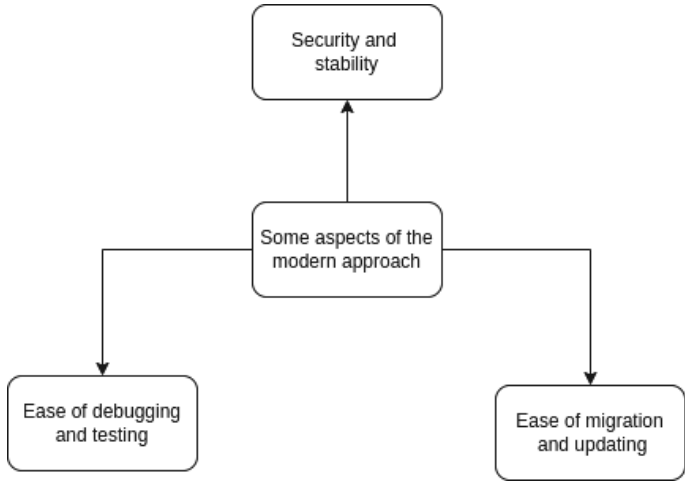


Figure 8.13. Some aspects of the modern approach to process isolation in the block-chain network

Containerization allows nodes with different configurations to run on a single system, simplifying the testing of new features or protocols. If detected, errors can be isolated and fixed without affecting the network's functionality.

Containers are easily migrated between servers or cloud platforms, simplifying scaling and maintaining the network. In addition, software updates on nodes can be carried out without stopping them or performing complex reconfigurations, reducing the likelihood of downtime.

The advantages of container models include versatility, where containers can be deployed on various operating systems and platforms, which provides flexibility in choosing an infrastructure.

Using a container model reduces maintenance costs due to the optimal use of computing resources and simplified management. This is reflected in the cost-effectiveness of building distributed systems based on blockchain.

Blockchain-based systems have an increased fault tolerance, as orchestrators automatically restart nodes in the event of a failure, ensuring the stability of the network.

Thus, container models are actively used in public and enterprise blockchain networks like Hyperledger Fabric, Ethereum, and Solana. In enterprise solutions, containerization allows you to create isolated test environments for testing smart contracts, develop and test new protocols, or provide flexible network management to support large-scale distributed systems.

Therefore, containerization in blockchain networks provides robust scalability, isolation, and network management tools. Containerization technologies, such as Docker, significantly simplify the deployment and maintenance of nodes, which helps to increase the efficiency and reliability of modern blockchain systems.

Distributed multi-node blockchain network models provide high flexibility and efficiency due to the use of modern node management and deployment technologies (Fig. 14). These models support scalability and simplify management due to integration with DevOps tools, high node density on a single server, and the ability to migrate or update the configuration quickly. Let's consider these characteristics in more detail.

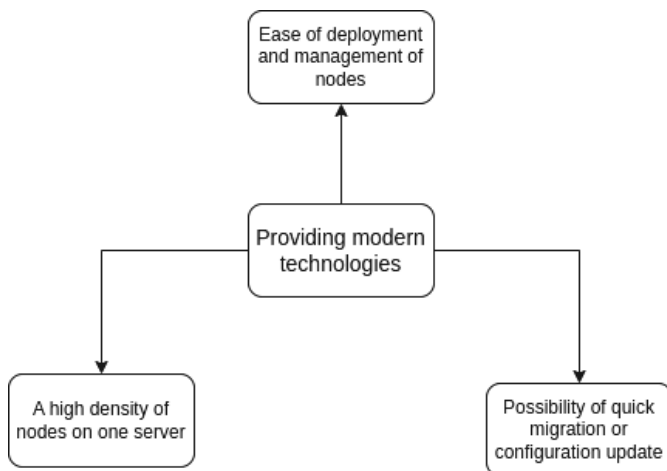


Figure 8.14. Using modern technologies for node management and deployment in blockchain network models

One key advantage of distributed multi-node models is the ability to use DevOps approaches to automate deploying, managing, and monitoring nodes. Tools such as Ansible, Terraform, and Kubernetes allow you to effectively manage the blockchain network infrastructure by automatically creating, configuring, and updating nodes.

Container management systems such as Docker and Docker Compose simplify the deployment process, minimizing manual intervention and reducing the likelihood of errors. Orchestrators like Kubernetes provide automatic scaling and resource management, especially important for highly loaded blockchain networks. This makes DevOps methodologies indispensable for large projects that require flexibility, reliability, and minimal time to deploy new nodes or change the network configuration.

Distributed multi-node models allow you to run multiple nodes on a single physical server thanks to virtualization and containerization technologies. Each node operates in an isolated environment, which prevents conflicts between processes and ensures their independence. This makes it possible to use the available server resources as efficiently as possible. High node density reduces hardware costs because a single server can host dozens of nodes without sacrificing performance. This approach is ideal for testing, developing, and operating private blockchain networks where balancing performance and resource savings is essential. Multi-node models provide a high degree of flexibility when changing infrastructure using containerization and virtualization standards. Nodes running in containers can be easily migrated between servers or cloud platforms, which speeds upscaling or failure recovery.

Orchestration technologies enable smooth configuration updates, eliminating the need to stop the network. This is achieved through rolling update strategies, in which nodes are updated in stages, ensuring continuous network availability. This approach minimizes downtime and reduces the risks of changing settings or software.

Thus, distributed multi-node blockchain network models have several advantages, including ease of management using DevOps tools, the possibility of high node density on a single server, and fast migration or configuration updates. These characteristics make them indispensable for implementing scalable, reliable, cost-effective blockchain systems in corporate and public networks.

Distributed multi-node blockchain network models are adequate for performing various tasks, including testing complex distributed systems, educational initiatives, and developing corporate blockchain solutions (Fig. 8.15).

Their architecture provides the flexibility, scalability, and isolation necessary for experimentation and implementing practical applications. Distributed multi-node models allow you to emulate the operation of complex blockchain networks with many nodes interacting through local or global networks. This makes it possible to study the behavior of the network in various scenarios, including high load, node failures, and attacks such as 51 % or DDoS attacks.

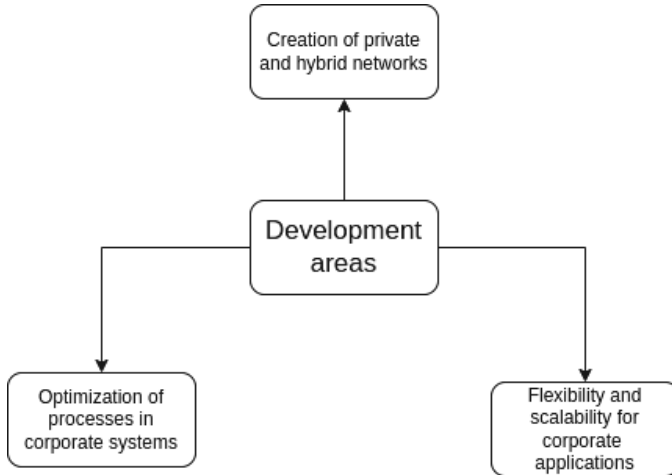


Figure 8.15. Development of corporate blockchain solutions

Testing complex distributed systems involves assessing the network performance with an increase in the number of nodes and transaction volume. For example, you can study the effect of the geographical distribution of nodes on data transfer delays and transaction confirmation times. Such a model allows you to identify bottlenecks and optimize consensus mechanisms and data routing.

Distributed multi-node architecture provides a secure environment for testing new consensus algorithms such as *Proof of Stake (PoS)* or *Delegated Proof of Stake (DPoS)*. This allows you to study their impact on security, transaction processing speed, and fault tolerance.

Distributed multi-node models are widely used in educational programs to teach students and professionals how to work with blockchain technologies. Creating and setting up a multi-node network provides an opportunity to learn key aspects of the blockchain ecosystem in practice, including validation processes, data synchronization, and smart contract management.

Their architecture provides the flexibility, scalability, and isolation necessary for experimentation and implementing practical applications. Distributed multi-node models allow you to emulate the operation of complex blockchain networks with many nodes interacting through local or global networks. This makes it possible to study the behavior of the network in various scenarios, including high load, node failures, and attacks such as 51 % or DDoS attacks.

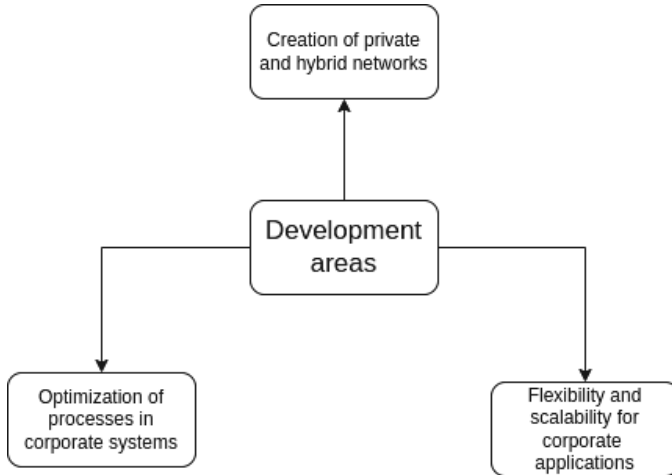


Figure 8.15. Development of corporate blockchain solutions

Testing complex distributed systems involves assessing the network performance with an increase in the number of nodes and transaction volume. For example, you can study the effect of the geographical distribution of nodes on data transfer delays and transaction confirmation times. Such a model allows you to identify bottlenecks and optimize consensus mechanisms and data routing.

Distributed multi-node architecture provides a secure environment for testing new consensus algorithms such as *Proof of Stake (PoS)* or *Delegated Proof of Stake (DPoS)*. This allows you to study their impact on security, transaction processing speed, and fault tolerance.

Distributed multi-node models are widely used in educational programs to teach students and professionals how to work with blockchain technologies. Creating and setting up a multi-node network provides an opportunity to learn key aspects of the blockchain ecosystem in practice, including validation processes, data synchronization, and smart contract management.

one can determine both the theoretical efficiency and practical applicability of architectures in blockchain networks.

8.6.2. Justification for the need for notations

Research and development of blockchain networks require a clear understanding of their architectural features. Notations perform several essential functions. Let's describe them.

Unification of description. Notations allow standardizing the description of architectural models. For example, instead of a detailed textual explanation of the characteristics of a local single-node structure, it is enough to use the *SL/BL/DN* notation, which concisely conveys the main parameters of the system.

Comparison of models. Using notations, it becomes possible to compare different architectures quickly. For example, you can instantly see that the *SH/BH/DH* model (distributed multi-node structure) outperforms *SL/BL/DN* (local single-node structure) in all key parameters.

Assessment of suitability for tasks. Notations allow you to assess whether a given model is suitable for solving a specific task. For example, a model with the *BH* notation is preferable in terms of security for an application that requires high security.

Ensuring academic rigor. The notational approach meets the requirements of scientific methodology, as it ensures the accuracy and unambiguity of terminology, which is essential for the reproducibility of research.

8.6.3. Benefits of using notations in scientific research

Here are some examples of the benefits of using notations in scientific research, which are as follows:

Conciseness. Notations reduce the volume of text descriptions, facilitating the perception and analysis of data.

Possibility of automation. Notations can be automatically used in software systems to select an architecture depending on user requirements.

Universality. The proposed approach applies to various types of blockchain networks, including corporate and public blockchains.

Support for comparative studies. Using notations, studies can be conducted to compare the performance, security, and decentralization of various architectures.

8.6.4. Examples of Blockchain Network Architectural Model Notations

An example of SL/BL/DN notation analysis.

This is a local single-node structure that has the following characteristics.

Scalability (S): low since a single node processes all transactions, limiting throughput.

Security (B): low since a node failure or hack will lead to complete network inaccessibility.

Decentralization (D): zero since there is no distribution.

These parameters are recorded in the SL/BL/DN notation, where:

S — Low (low scalability);

B — Low (low security);

D — None (no decentralization).

The role of the notation in this case is presented as follows.

Systematization. The SL/BL/DN notation systematizes the characteristics of the architecture, allowing it to be easily identified.

Disadvantage analysis. The local single-node structure is unsuitable for high-security applications and throughput requirements.

Optimization of solutions. The notation helps choose a more appropriate model, such as switching to a local multi-node structure SM/BM/DL.

Thus, notations such as SL/BL/DN are powerful tools for studying and classifying architectural models of blockchain networks. They help systematize knowledge and open new opportunities for analyzing and optimizing existing solutions. Academic application of this approach will allow the development of blockchain technologies in a more structured and efficient manner.

An example of analyzing the SM/BM/DL notation.

This is a local multi-node structure that has the following characteristics.

Scalability (S): medium, since several nodes can process transactions in parallel, increasing throughput.

Security (B): medium, since if one node fails, the others continue functioning, ensuring network stability.

Decentralization (D) is low since all nodes are located on one server, limiting the distribution level.

These parameters are fixed in the SM/BM/DL notation, where:

S — Medium (medium scalability);

B — Medium (medium security);

D — Low (low decentralization).

The role of the notation in this case is presented as follows:

Systematization. The *SM/BM/DL* notation allows you to structure the main characteristics of the local multi-node architecture, providing its unambiguous description.

Disadvantage analysis. The notation demonstrates that this structure's level of decentralization is limited, making it less suitable for tasks that require high resistance to external attacks.

Optimization of solutions. The use of the notation helps to identify areas for improvement. For example, to increase the level of decentralization, you can switch to a distributed single-node *SL/BM/DM* structure or a distributed multi-node *SH/BH/DH* structure.

Thus, notations like *SM/BM/DL* effectively describe and analyze local multi-node blockchain network architectures. This model is characterized by medium scalability and security, which makes it optimal for testing consensus algorithms and modeling network interactions. Despite the low degree of decentralization, the notation allows you to clearly define the model's limitations and its potential in an isolated environment. Using this approach in scientific research contributes to the standardization of analysis methods, improving the quality and systematicity of the study of architectural solutions of blockchain networks.

An example of SL/BM/DM notation analysis.

This is a distributed single-node structure that has the following characteristics.

Scalability (S): low since a single node processes all transactions, which limits throughput.

Security (B): medium, since placing a node in a geographically distributed network reduces the likelihood of a complete loss of functionality.

Decentralization (D): medium, since the node is accessible for network interaction, ensuring basic distribution.

These parameters are captured in the *SL/BM/DM* notation, where:

S — Low (low scalability);

B — Medium (medium security);

D — Medium (medium decentralization).

The role of the notation in this case is presented as follows.

Systematization. The *SL/BM/DM* notation systematizes the characteristics of a distributed single-node architecture, facilitating its identification and description.

Disadvantage analysis. The notation demonstrates that this structure's main limitation is low scalability, which makes it less practical for applications with high throughput requirements.

Optimization of solutions. Using the notation allows for the identification of areas for improvement. For example, switching to a distributed multi-node *SH/BH/DH* structure to improve scalability and decentralization.

Thus, notations such as *SL/BM/DM* provide a structured approach to analyzing a distributed single-node architecture of blockchain networks. This model has low scalability but provides a medium level of security and decentralization due to the node's placement in a geographically distributed network. The notation allows one to accurately describe this architecture's strengths and weaknesses, making it a valuable tool for assessing its suitability in resource-constrained conditions or demonstrating interaction in a distributed environment. The notational approach contributes to further developing research methods and optimizing blockchain solutions.

An example of analyzing the SH/BH/DH notation.

This is a distributed multi-node structure that has the following characteristics.

Scalability (S): high since the nodes process transactions in parallel, significantly increasing the throughput.

Security (B) is high, as hacking one node does not disrupt the entire network, thanks to the decentralized structure and data replication.

Decentralization (D) is high because the nodes are distributed worldwide, ensuring uniform load distribution and resistance to external threats.

These parameters are recorded in the *SH/BH/DH* notation, where:

S — High (high scalability);

B — High (high security);

D — High (high decentralization).

The role of the notation in this case is presented as follows.

Systematization. The *SH/BH/DH* notation systematizes the characteristics of a distributed multi-node architecture, allowing for an adequate description of its key properties and potential applications.

Analysis of advantages. The notation emphasizes that this structure is optimal for applications that require high performance, fault tolerance, and complete decentralization, such as public blockchain networks (e.g., Ethereum and Bitcoin).

Optimization of solutions. Notations allow us to highlight the architecture's strengths and consider its adaptation for specific tasks, such as scalable enterprise blockchain solutions or the creation of sustainable decentralized financial platforms (*DeFi*).

Thus, notations like *SH/BH/DH* are essential for analyzing and classifying blockchain networks' distributed multi-node architecture. This model

provides high scalability, security, and decentralization, making it optimal for public blockchain networks with high performance and sustainability requirements.

The notations discussed above allow us to clearly and concisely describe this architecture's key advantages, facilitating its use in scalable and fault-tolerant systems. Academic implementation of this approach helps structure research and develop innovative solutions in blockchain technologies.

8.7. BLOCKCHAIN NETWORK ARCHITECTURAL MODEL NOTATIONS AND THE BLOCKCHAIN TRILEMMA

The blockchain trilemma remains a fundamental challenge in the development of blockchain systems. Despite significant progress in understanding and formalizing the problem, a definitive solution that simultaneously achieves high decentralization, security, and scalability has yet to be found. Current research continues to propose innovative approaches and architectures to mitigate the trilemma's limitations, but each faces its tradeoffs and practical limitations. Below, we review some of the views on the blockchain trilemma in recent years.

For example, Massassi and El Kalam [21] recently published a formal mathematical proof of the blockchain trilemma. They demonstrated that maximizing decentralization, security, and scalability simultaneously leads to internal inconsistencies. Their analysis includes a contradiction proof and computational complexity analysis, deepening the understanding of blockchain technology's fundamental limitations.

Fu et al. [22] conducted a comparative analysis of the Algorand and Ethereum 2.0 platforms, evaluating them according to the criteria of the blockchain trilemma. Their study interprets existing indices for measuring decentralization, evaluates scalability through transaction data, and analyzes security to identify potential vulnerabilities. The results highlight the strengths of each platform and propose general methodologies for assessing key blockchain characteristics that apply to other systems.

Del Monte et al. [23] theoretically described a new blockchain architecture that scales to arbitrarily high loads given a proportional increase in the number of nodes. They argue that their approach does not require tradeoffs in security or decentralization, thereby challenging the traditional understanding of the trilemma.

Another paper [24], presented at IEEE, developed a framework for evaluating and comparing third-generation blockchain systems based on three core non-functional aspects: scalability, decentralization, and security. This framework provides validated methods for quantifying the tradeoffs associated with the trilemma and can serve as a guide for the design and orchestration of complex blockchain-based service applications.

The blockchain trilemma is a concept first formulated to explain the problems that developers face when creating blockchain networks. According to this concept, it is impossible to simultaneously optimize three key parameters: scalability, security, and decentralization. This limitation significantly impacts the design of blockchain architectural models and determines the choice of suitable solutions for various application areas.

The blockchain trilemma concept is to define:

- scalability as the network’s ability to process a large number of transactions per unit of time;
- security as the resistance of the blockchain network and nodes to attacks, including hacking of nodes, data substitution, and system failures;

Decentralization is the equal distribution of rights and responsibilities among network participants, excluding the dominance of one or more nodes.

Each described blockchain architecture chooses a compromise between these characteristics, expressed in its strengths and weaknesses. Let’s consider the relationship of the trilemma with blockchain network architectural models.

Local single-node architecture (*SL/BL/DN*) sacrifices security and decentralization to simplify deployment and minimize costs. It is suitable only for isolated environments, such as testing.

Local multi-node architecture (*SM/BM/DL*) involves a trade-off: It increases scalability and security but sacrifices decentralization. This architecture is helpful for network modeling or testing consensus algorithms.

Due to its geographic distribution, distributed single-node architecture (*SL/BM/DM*) improves security and decentralization but is limited by low throughput. Its application is relevant for demonstrating the operation of blockchain technologies and analyzing network latency.

Distributed multi-node architecture (*SH/BH/DH*) is the most balanced but requires significant resources to ensure high scalability, security, and decentralization. It is used in public blockchain networks such as Bitcoin and Ethereum.

8.8. USING NOTATIONS FOR MODELING BLOCKCHAIN PROCESSES

Blockchain technology has emerged as a transformative force across various industries, offering decentralized, secure, and transparent solutions for data management and transactions. In Business Process Management (BPM), blockchain integration presents unique opportunities and challenges, particularly concerning modeling blockchain processes. This literature review examines recent scholarly contributions to modeling blockchain processes within BPM, focusing on sources published in the last 5 to 10 years.

Taherdoost and Madanchian [25] comprehensively analyzed modeling approaches integrating blockchain with BPM. Their study highlights blockchain's potential to enhance business processes' transparency, security, and efficiency. They discuss traditional BPM models, blockchain-oriented models, and hybrid approaches, emphasizing the need for adaptable modeling techniques to incorporate blockchain's decentralized nature effectively.

Mohammed et al. [26] performed a systematic literature review focusing on blockchain's revolutionary impact on modern business. Their research identifies key themes and advancements in blockchain adoption, noting a shift towards environmentally sustainable supply chains and virtual environments. The study underscores the importance of developing conceptual frameworks to analyze blockchain's integration into business processes.

Curty et al. [27] explored model-driven engineering (MDE) and low-code/no-code platforms in designing blockchain-based applications. Their structured literature review reveals that MDE approaches can significantly reduce the technical complexity of blockchain application development, making it more accessible to a broader audience. The study also identifies opportunities for future research to enhance these modeling techniques.

Stiehle and Weber [28] developed a taxonomy through a systematic literature review to facilitate the implementation of inter-organizational business processes using blockchain. Their research addresses challenges such as traceability, correctness, flexibility, and scalability in blockchain-based process enactment, providing a structured framework for future studies.

Six et al. [29] conducted a systematic literature review to identify software patterns pertinent to the design of decentralized blockchain applications. They compiled a collection of 120 unique blockchain-related patterns and proposed a taxonomy to assist software architects in selecting appropriate patterns for their applications, ensuring software quality and reliability.

Labazova et al. [30] proposed a process model for configuring blockchain systems and applications, bridging knowledge gaps between comput-

er science and socio-economic research. Their model assists in selecting and configuring blockchain systems based on specific project requirements, contributing to more effective management of blockchain applications.

Kupke and Trattnig [31] introduced a systematic method for decentralizing centralized process models extended with blockchain annotations. Their approach involves creating individual local process models for each organization and a separate model for the blockchain, facilitating the implementation of collaborative processes using blockchain technology.

A study published in IEEE [32] examines how blockchain technology can enhance collaborative BPM by promoting decentralized architecture and secure data sharing among entities. The research highlights blockchain's potential to improve trust and efficiency in collaborative business processes.

Schmidt and Wagner [33] investigated how blockchain technology influences business models. Their study provides a taxonomy of business models that utilize blockchain, offering insights into how organizations can adapt their processes and structures to leverage blockchain's capabilities effectively.

Thus, integrating blockchain technology into business process modeling is a dynamic and evolving field. The reviewed literature underscores the necessity for adaptable and robust modeling approaches to accommodate blockchain's decentralized and immutable characteristics. Future research should explore hybrid models and develop comprehensive frameworks to guide the effective implementation of blockchain in BPM.

The notations proposed for classifying architectural models of blockchain networks (*SL/BL/DN*, *SM/BM/DL*, *SL/BM/DM*, *SH/BH/DH*) may be necessary for modeling processes in blockchain networks. They allow for formalizing the characteristics of the architecture and making the analysis, development, and optimization of blockchain networks more efficient. Let's consider the main areas of their application.

8.8.1. Formalization of architectural solutions

Notations facilitate the creation of mathematical models of the blockchain. For example, the *SL/BL/DN* notation is a simple model in which all processes are concentrated on one node. This allows you to study performance with limited resources.

The *SH/BH/DH* notation is a more complex decentralized model that models the interaction processes between multiple nodes, network delays, and consensus.

Notations such as $SL/BL/DN$ and $SH/BH/DH$ allow you to formalize the key aspects of architectural models of blockchain networks, which significantly facilitates their mathematical modeling and subsequent analysis (tabl. 8.3). Let's take a closer look at how this works in practice.

A local single-node structure in the form of $SL/BL/DN$ notation describes an architecture where all processes are executed on a single node. The node processes transactions, stores data, and provides all the necessary blockchain functionality. Mathematical modeling of this notation will consist of studying the performance of a system that uses the $M/M/1$ queuing system. Moreover, M will be a transaction flow that obeys the *Poisson distribution*. The following parameter, 1, represents a system with one server (node) processing transactions.

The formula for the average delay will be:

$$W = \frac{1}{\mu - \lambda},$$

where

μ is the speed of transaction processing by the node,

λ is the intensity of transaction receipt.

The practical application of the $SL/BL/DN$ notation will consist of:

- modeling a local network for testing basic operations, such as executing smart contracts or processing transactions;
- analyzing performance with limited resources, for example, in low-budget systems.

A distributed multi-node structure, in the form of $SH/BH/DH$ notation, is a decentralized architecture in which many nodes interact. Each node processes transactions in the network in parallel concerning the neighboring node, participates in the consensus, and stores data in a distributed manner.

Mathematical modeling of this notation will consist of studying the processes of interaction between nodes, where the graph model is used as follows:

- nodes (N) are graph vertices;
- graph edges model communication channels between nodes with a given throughput (B_{ij}).

Modeling of the consensus of the blockchain network can be represented as a process of synchronization of the states of nodes:

$$S_i(t+1) = f(S_i(t), S_j(t), \dots),$$

where

$S_i(t)$ is the state of node i at time t , and f is the consensus function (e.g., *PBFT* or *PoS*).

The $M/G/I$ queue model can be used to model and analyze network delays between nodes, where G is the total network delay, including random components (e.g., packet loss, routing).

Network throughput is calculated as a function of the number of nodes (N), average latency (D), and the number of transactions (T) as follows:

$$TPS = \frac{T}{N \cdot D}.$$

The practical application of the transactions considered is as follows:

- analyzing the performance of large public blockchain networks such as Ethereum;
- modeling the impact of network latency on consensus time;
- optimizing the network structure to reduce transaction processing time.

Table 8.3

Comparative characteristics of $SL/BL/DN$ and $SH/BH/DH$ notations for modeling processes in a blockchain network

Characteristic	$SL/BL/DN$	$SH/BH/DH$
Scalability	Low since one node limits the transaction processing speed (TPS).	High since nodes process transactions in a blockchain network in parallel.
Modeling complexity	Low, since the general modeling scheme is a simple structure.	High, since the general modeling scheme requires additional consideration of node interactions.
Application	Local test networks.	Public decentralized networks.

Thus, the role of notations in modeling is to adapt to various tasks in blockchain networks. For example, in a project with limited resources, the $SL/BL/DN$ notation can be used for preliminary analysis of the blockchain network. In more complex distributed systems where high performance is required, models corresponding to $SH/BH/DH$ are used. At the same time, notations such as $SL/BL/DN$ and $SH/BH/DH$ help formalize the system's key parameters, making the modeling process more structured and convenient for analysis. Therefore, formalized notations make the modeling results easily reproducible and comparable between researchers.

8.8.2. Practical Use of Blockchain Network Simulation Tools

Simulation tools are essential for analyzing, testing, and optimizing blockchain networks. They allow you to simulate complex interactions between nodes, transactions, and consensus algorithms, which is especially useful for scientific research and developing new solutions. Let's consider the key aspects of using such simulation tools, which involve obtaining experimental study results (some key parameters). To do this, scenarios of load effects on one or more blockchain network nodes are created to obtain key parameters. Such key parameters for any notation may include the following:

- throughput to estimate the speed of transaction processing per second:

$$TPS = \frac{T_{total}}{t_{end} - t_{start}},$$

where

T_{total} is the total number of confirmed transactions,
 t_{start} and t_{end} are the beginning and end of the test.

- transaction processing delay to estimate the average time from the moment a transaction is sent to its confirmation:

$$Latency = \frac{\sum_{i=1}^N (t_{confirmed} - t_{submitted})}{N},$$

where

$t_{confirmed}$ and $t_{submitted}$ are the time of confirmation and sending of the transaction, respectively;

- resource usage is used to determine estimates for monitoring network nodes' use of processor, memory, and network resources.

Next, we will consider examples of research scenarios based on testing blockchain network nodes.

1. Ethereum network throughput assessment scenario.

Description. Testing the network for the number of transactions it can process per unit of time.

Test parameters:

- network type: private Ethereum network (based on Geth);
- block size: configured for testing;
- Transaction flow is uniform and has increasing intensity.

Objective:

- 1) measure the maximum *TPS* with increasing load;
- 2) determine the impact of block size on performance.

Research procedure:

- 1) Launch the Ethereum network using Geth.
- 2) Configure the software tool for sending transactions (scenario 8.1).

```
{
  "test": {
    "name": "TPS Test",
    "description": "Measure transaction throughput",
    "workers": [
      {
        "rateControl": {
          "type": "fixed-rate",
          "opts": {
            "tps": 50
          }
        },
        "rounds": [
          {
            "transaction": "simple",
            "txDuration": 60
          }
        ]
      }
    ]
  }
}
```

Scenario 8.1. Ethereum network throughput assessment

- 3) Run the test and analyze the results.

Expected results:

- 1) determine the maximum TPS of the network;
- 2) determine the delay between sending and confirming transactions.

2. Smart contract testing scenario.

Description. Network performance assessment during mass execution of smart contracts.

Test parameters:

- smart contract: ERC-20 conditional token;
- number of functions: 3 (create, transfer, balance check);
- transaction flow: 100 transactions per second.

Objective:

- 1) assess the execution time of smart contract functions;

2) determine the network load during the mass use of smart contracts.

Research procedure:

1) deploy an ERC-20 smart contract on the Ethereum network.

2) configure a scenario (scenario 8.2) for interaction with the smart contract:

```
{
  "test": {
    "name": "Smart Contract Test",
    "description": "Measure performance of ERC-20 contract",
    "workers": [
      {
        "rateControl": {
          "type": "fixed-rate",
          "opts": {
            "tps": 100
          }
        },
        "rounds": [
          {
            "transaction": "invoke",
            "arguments": {
              "function": "transfer",
              "args": ["0x1234...", "10"]
            },
            "txDuration": 60
          }
        ]
      }
    ]
  }
}
```

Scenario 8.2. Smart contract performance assessment

3) run tests for the transfer, approve, and balanceOf functions.

Expected results:

1) average execution time of each function;

2) node load level during smart contract execution.

3. *Scenario for testing the scalability of a blockchain network.*

Description. Studying the impact of increasing the number of nodes in a network on its performance.

Test parameters:

- number of nodes: 5, 10, 20;
- network type: Private network based on Ethereum;
- transaction type: Simple Ether transfers.

Objective:

- 1) measure how increasing the number of nodes affects TPS;
- 2) evaluate delays in distributing blocks across the network.

Research procedure:

- 1) set up an Ethereum network with a variable number of nodes;
- 2) use tools for sending transactions with a uniform load (scenario 8.3);

```
{
  "test": {
    "name": "Scalability Test",
    "description": "Measure scalability with increasing nodes",
    "workers": [
      {
        "rateControl": {
          "type": "fixed-rate",
          "opts": {
            "tps": 200
          }
        },
        "rounds": [
          {
            "transaction": "simple",
            "txDuration": 120
          }
        ]
      }
    ]
  }
}
```

Scenario 8.3. Evaluation of the scalability of a blockchain network

- 3) perform testing for each configuration option of the network.

Expected results:

- 1) determine the change in *TPS* with an increase in the number of nodes;
- 2) determine the time to reach a consensus.

Thus, some tools effectively determine the performance of Ethereum networks in various scenarios. They allow you to identify bottlenecks, op-

timize network parameters, and ensure stable operation under high load conditions. Due to their flexibility and wide range of functions, these tools are indispensable for research and development in blockchain technologies.

8.9. DISCUSSION

Researching the specified models and their notations allows us to determine the optimal architecture models for various areas of blockchain technology application. This contributes to the development of technologies by increasing their sustainability, performance, and security. In addition, each study provides new opportunities for adapting the blockchain to users' unique requirements.

The proposed notational system allows us to unify the description of architectural models of blockchain networks and conduct their comparative analysis. For example, a local single-node structure (*SL/BL/DN*) is suitable for testing. Still, it does not meet the requirements of security and decentralization, while a distributed multi-node structure (*SH/BH/DH*) is optimal for tasks requiring high fault tolerance and scalability.

This paper considers the blockchain trilemma, which emphasizes the fundamental limitations in designing blockchain network architectures. Each model solves this problem: local structures sacrifice decentralization, distributed single-node networks limit scalability, and distributed multi-node architectures require more resources to balance parameters. Analyzing the trilemma in the context of architecture classification allows us to understand the trade-offs better and adapt blockchain solutions to specific tasks.

8.10. CONCLUSION

Studying architectural models of blockchain networks is necessary due to their widespread use and high importance in modern technologies. Developing a notational system for describing architectural models of blockchain networks is essential for analyzing their characteristics and choosing an appropriate architecture. A deep understanding of each model allows for improving existing solutions and opening new horizons for developing decentralized technologies.

Thus, systematic research in this area is the foundation for successfully integrating blockchain networks into various fields of activity. The concept

of the blockchain trilemma is a key factor determining the design and development of blockchain network architectures. The relationship of the trilemma with architectural models such as *SL/BL/DN*, *SM/BM/DL*, *SL/BM/DM*, and *SH/BH/DH* demonstrates the need for a compromise between scalability, security, and decentralization.

Using notations such as *SL/BL/DN* and *SH/BH/DH* allows for the efficient description of blockchain network architectures and their mathematical modeling. Simple models are applicable for local systems, while more complex notations are needed to analyze large decentralized networks. This approach facilitates the development, testing, and optimization of blockchain solutions.

The study of blockchain network architectures requires a strictly structured methodology that combines mathematical models, simulations, and specialized tools. This paper examines approaches to the analysis and modeling of architectural solutions, the methods used to evaluate their characteristics, and the capabilities and limitations of existing tools. The need for an integrated approach, including theoretical and practical methods, is substantiated to obtain reliable results and develop effective blockchain architectures.

Future research should aim to overcome some of the trilemma's limitations. This will allow for the creation more efficient and versatile blockchain solutions for a given task.

REFERENCES

1. Fu X, Wang H, Shi P. A survey on blockchain architectures: taxonomies, consensus mechanisms, and challenges. *Sci China Inf Sci*. 2021;64:121101. <https://doi.org/10.1007/s11432-019-2790-1>.
2. Singh A, Kumar G, Saha R, et al. A survey and taxonomy of consensus protocols for blockchains. *J Syst Archit*. 2020;110:101861. <https://doi.org/10.1016/j.sysarc.2020.101861>.
3. Islam S, Islam MJ, Hossain M, et al. A survey on consensus algorithms in blockchain-based applications: architecture, taxonomy, and operational issues. *IEEE Access*. 2023. <https://doi.org/10.1109/ACCESS.2023.10101789>.
4. Uriarte RB, De Nicola R. Blockchain-based decentralized cloud/fog solutions: challenges, opportunities, and standards. *IEEE Commun Standards Mag*. 2018. <https://doi.org/10.1109/MCOMSTD.2018.8515145>.
5. Zheng Z, Xie S, Dai H, et al. An overview of blockchain technology: architecture, consensus, and future trends. In: *Proc IEEE Int Congr Big Data*. 2017. p. 557–64. <https://doi.org/10.1109/BigDataCongress.2017.85>.

6. Current trends of blockchain technology: architecture, applications, and future directions. *J Cloud Comput.* 2024. <https://doi.org/10.1007/s43926-024-00058-5>.
7. Gupta S, Sadoghi M. Blockchain transaction processing. In: Sakr S, Zomaya AY, editors. *Encyclopedia of big data technologies*. Berlin: Springer; 2019. p. 366–76. https://doi.org/10.1007/978-3-319-77525-8_333.
8. Peng L, Feng W, Yan Z, et al. Privacy preservation in permissionless blockchain: a survey. *Digit Commun Netw.* 2021;7(3):295–307. <https://doi.org/10.1016/j.dcan.2020.05.008>.
9. Christidis K, Devetsikiotis M. Blockchains and smart contracts for the internet of things. *IEEE Access.* 2016;4:2292–303. <https://doi.org/10.1109/ACCESS.2016.2566339>.
10. Xiao Y, Zhang N, Lou W, Hou YT. A survey of distributed consensus protocols for blockchain networks. *IEEE Commun Surveys Tuts.* 2020;22(2):1432–70. <https://doi.org/10.1109/COMST.2020.2969706>.
11. Wang W, Hoang DT, Hu P, et al. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access.* 2019;7:22328–57. <https://doi.org/10.1109/ACCESS.2019.2896108>.
12. Zamyatin A, Al-Bassam M, Meiklejohn S, et al. Sok: communication across distributed ledgers. *IEEE Commun Surveys Tuts.* 2021;23(2):954–97. <https://doi.org/10.1109/COMST.2021.3050157>.
13. Zhang P, White J, Schmidt DC, et al. A survey on blockchain-based systems and applications for healthcare. *IEEE Access.* 2018;6:9441–73. <https://doi.org/10.1109/ACCESS.2018.2792944>.
14. Nguyen QK. Blockchain: a data structure for maintaining trustworthy records. In: 2016 IEEE Int Conf Collaboration Internet Computing. IEEE; 2016. p. 490–4. <https://doi.org/10.1109/CIC.2016.078>.
15. Ali ST, Nelson J, Shea R. Blockstack: a global naming and storage system secured by blockchains. In: *Proc USENIX Symp Networked Syst Design Implementation*. 2016. p. 181–196. <https://doi.org/10.5555/2930611.2930631>.
16. Wood G. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum White Paper*. 2014. <https://ethereum.org/en/whitepaper/>.
17. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2008. <https://bitcoin.org/bitcoin.pdf>.
18. Buterin V. A next-generation smart contract and decentralized application platform. *Ethereum White Paper*. 2013. <https://ethereum.org/en/whitepaper/>.
19. Back A, Corallo M, Dashjr L, et al. Enabling blockchain innovations with pegged sidechains. *Blockstream White Paper*. 2014. <https://blockstream.com/sidechains.pdf>.
20. Bonneau J, Miller A, Clark J, et al. Sok: research perspectives and challenges for bitcoin and cryptocurrencies. In: *Proc IEEE Symp Security Privacy*. 2015. p. 104–121. <https://doi.org/10.1109/SP.2015.14>.

21. Mssassi S, El Kalam AA. The Blockchain Trilemma: A Formal Proof of the Inherent Trade-Offs Among Decentralization, Security, and Scalability. *Appl. Sci.* 2025;15(1):19. <https://doi.org/10.3390/app15010019>.
22. Fu Y, Jing M, Zhou J, et al. Quantifying the Blockchain Trilemma: A Comparative Analysis of Algorand, Ethereum 2.0, and Beyond. *arXiv preprint arXiv:2407.14335*. 2024. <https://arxiv.org/abs/2407.14335>.
23. Del Monte G, Pennino D, Pizzonia M. Scaling Blockchains Without Giving up Decentralization and Security. *arXiv preprint arXiv:2005.06665*. 2020. <https://arxiv.org/abs/2005.06665>.
24. The Blockchain Trilemma: An Evaluation Framework. *IEEE Xplore*. 2024. <https://ieeexplore.ieee.org/document/10572260>.
25. Taherdoost, H., & Madanchian, M. (2024). Blockchain and Business Process Management (BPM) Synergy: A Comparative Analysis of Modeling Approaches. *Information*, 15(1), 9. <https://doi.org/10.3390/info15010009>
26. Mohammed, M. A., De-Pablos-Heredero, C., & Montes Botella, J. L. (2024). A Systematic Literature Review on the Revolutionary Impact of Blockchain in Modern Business. *Applied Sciences*, 14(23), 11077. <https://doi.org/10.3390/app142311077>
27. Curty, S., Hdrer, F., & Fill, H.-G. (2023). Design of Blockchain-Based Applications Using Model-Driven Engineering and Low-Code/No-Code Platforms: A Structured Literature Review. *Software and Systems Modeling*, 22, 1857–1895. <https://doi.org/10.1007/s10270-023-01109-1>
28. Stiehle, F., & Weber, I. (2022). Blockchain for Business Process Enactment: A Taxonomy and Systematic Literature Review. *arXiv preprint arXiv:2206.03237*. <https://doi.org/10.48550/arXiv.2206.03237>
29. Six, N., Herbaut, N., & Salinesi, C. (2022). Blockchain Software Patterns for the Design of Decentralized Applications: A Systematic Literature Review. *arXiv preprint arXiv:2201.04374*. <https://doi.org/10.48550/arXiv.2201.04374>
30. Labazova, O., Kazan, E., Dehling, T., Tuunanen, T., & Sunyaev, A. (2021). Managing Blockchain Systems and Applications: A Process Model for Blockchain Configurations. *arXiv preprint arXiv:2105.02118*. <https://doi.org/10.48550/arXiv.2105.02118>
31. Kipke, J., & Trattning, L. (2023). Transformation Rules for Decentralizing Blockchain-Extended Process Models. *Business & Information Systems Engineering*, 65(2), 103–118. <https://doi.org/10.1007/s12599-022-00753-8>
32. IEEE Standards Association. (2020). Using Blockchain to Improve Collaborative Business Process Management. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(11), 6429–6441. <https://doi.org/10.1109/TSMC.2020.2999432>
33. Schmidt, C. G., & Wagner, S. M. (2019). Blockchain and Supply Chain Relations: A Transaction Cost Theory Perspective. *Journal of Business Logistics*, 40(4), 255–276. <https://doi.org/10.1111/jbl.12241>