Scalability and Security in Blockchain Networks: Evaluation of Sharding Algorithms and Prospects for Decentralized Data Storage

Andrey L. Bulgakov , Anna V. Aleshina * , Sergey D. Smirnov , Alexey D. Demidov , Maxim A. Milyutin , Yanliang Xin

Posted Date: 14 October 2024

doi: 10.20944/preprints202410.1078.v1

Keywords: blockchain technologies; sharding; scalability; security; green bonds; sustainable development; decentralization

Article

Scalability and Security in Blockchain Networks: Evaluation of Sharding Algorithms and Prospects for Decentralized Data Storage

Andrey L. Bulgakov [1,2], Anna V. Aleshina [1,2,*], Sergey D. Smirnov [1], Alexey D. Demidov [1], Maxim A. Milyutin [1] and Xin Yanliang [2]

[1] Moscow Center for Fundamental and Applied Mathematics, Lomonosov Moscow State University, 119991, Russian Federation
[2] Faculty of Economics, Lomonosov Moscow State University, 119991, Russian Federation
* Correspondence: bulgakoval@my.msu.ru; aleshina@econ.msu.ru; smirbox@gmail.com; demidov.ad@bk.ru; milyutinma@my.msu.ru; xinyanliang@yandex.com

Abstract: The article addresses the issues of scalability and security in blockchain networks, with a focus on sharding algorithms and decentralized data storage. Key challenges include the low throughput and high transaction latency in public networks such as Bitcoin and Ethereum. Sharding is examined as a method to enhance performance through data distribution, but it raises concerns regarding node management and reliability. Sharding schemes, such as Elastico, OmniLedger, Pyramid, RepChain, and SSchain, are analyzed, each presenting its own advantages and drawbacks. Alternative architectures like Directed Acyclic Graphs (DAG) demonstrate potential for improved scalability but require further refinement to ensure decentralization and security. Protocols such as Brokerchain, Meepo, AHL, Benzene, and CycLedger offer unique approaches to addressing performance and transaction consistency issues. The article emphasizes the need for a comprehensive approach, including dynamic sharding, multi-level consensus, and inter-shard coordination. Additionally, a conceptual model is proposed that incorporates sharding of transactions, states, and networks, which enables greater scalability and efficiency.

Keywords: blockchain technologies; sharding; scalability; security; green bonds; sustainable development; decentralization

1. Introduction

In recent years, blockchain technologies have attracted considerable attention from both academia and industry, promising to revolutionize areas ranging from finance to supply chain management. One of the key aspects that ensures the successful adoption and widespread adoption of blockchains is their ability to scale and provide reliable decentralized data storage. However, despite their numerous advantages, blockchain networks face a number of serious challenges, with scalability and security issues occupying a central place. The scalability of blockchain networks is limited by their architectural features, such as low throughput and high transaction confirmation latency. This is especially true for first- and second-generation public blockchains such as Bitcoin and Ethereum, which are unable to cope with the growing number of users and transactions. The CAP theorem highlights the fundamental limitations of distributed systems, indicating the impossibility of simultaneously achieving consistency, availability, and fragmentation resistance. In this context, sharding appears to be one of the most promising methods for solving the scalability problem. Sharding is a method of data distribution that divides the system into smaller segments (shards), allowing transactions to be processed in parallel and improving overall network performance. There are several approaches to sharding, including key-based and directory-based methods, each with their own advantages and disadvantages. Protocols such as Benzene, CycLedger, and OmniLedger offer unique solutions to improve performance and security, but also face a number of challenges,

 doi:10.20944/preprints202410.1078.v1

including node management and ensuring data reliability. Additionally, alternative architectures such as directed acyclic graphs (DAGs) show potential for improving scalability by providing high speed and throughput without the need for mining. However, they also require improvements to ensure decentralization and fraud protection. The Segwit and MAST protocols implemented in the Bitcoin network are examples of successful solutions aimed at improving scalability and transaction efficiency.

The purpose of this article is to conduct a comprehensive assessment and analysis of existing sharding algorithms and other approaches to decentralized data storage in blockchain networks. The study aims to identify the strengths and weaknesses of various methods, as well as to determine the prospects for their further development. The introduction emphasizes the need for an integrated approach to solving scalability and security issues, making this study an important step towards sustainable development of blockchain technologies.

The active implementation of blockchain technologies has revealed its fundamental contradictions and shortcomings, the solution of which is a key task for modern IT leaders. We are talking mainly about the problems of scalability, performance and coexistence of the blockchain pillars that underlie its conceptual model and ultimately predetermined its victory over existing algorithms for the transfer and storage of information. At the end of the last century, Eric Brewer formulated and proved the CAP theorem (from the English consistency, availability, partition), stating that in a distributed system it is impossible to ensure a stable balance and combination between consistency, availability and resistance to fragmentation. A logical consequence of the theorem was the conclusion about the incompatibility of the fundamental principles of modern blockchain: decentralization, scalability, sustainability, trust, security, etc. In each specific situation, developers face a choice, the creation of any blockchain technology begins with a search for a compromise between its named attributes. Blockchains based on acyclic graphs achieve a high level of scalability and decentralization with questionable security, security and scalability are achieved in consortium blockchains and private blockchains, but at the expense of decentralization: all of them are fully or partially centralized. Public blockchains are well protected and decentralized, but have poor scalability (for example, Bitcoin, the scalability problem of which has not yet been solved due to the specifics of the Proof of Work consensus algorithm used in it, which is almost incompatible with sharding)

The newborn and immature nature of blockchain technology explains the absence of any meaningful and influential scientific school that would create a generally accepted terminology base and provide answers or at least recipes for finding them to the most pressing questions in the field of blockchain; in this series - solving the scalability problem, strengthening security, increasing network throughput, improving consensus algorithms and much more. Blockchain can be private, public, or a consortium, but its ideological structure is always the same and consists of five levels: application level, data level, consensus level, network level, and platform level. The platform level includes the final products of the blockchain, such as cryptocurrency. The data level consists of hashing algorithms, data creation and verification, and digital signatures. The consensus level regulates and governs consensus algorithms. The network level includes data compression and distribution algorithms, as well as proctologist and services for exchanging messages between network participants. The platform level is the blockchain infrastructure, its software and hardware system, and technological basis. Sustainable development of blockchain technology requires coordination and regulation of all the above levels.

This article is mainly devoted to an overview of existing approaches to sharding and their technological features, at the same time assessing them in the context of their effectiveness, feasibility of use and practical significance for solving scalability problems. To begin with, let's get rid of the free interpretations of the basic conceptual and categorical apparatus that we will have to use. Even authoritative publications often make a terminological inaccuracy, if not an outright mistake, identifying the concepts of sharding and partitioning. Let's start with the fact that sharding involves storing shards on different servers, in contrast to partitions, which are stored in most cases on one. In addition, partitioning can be both horizontal and vertical. Sharding can only be horizontal, its vertical

 doi:10.20944/preprints202410.1078.v1

modification does not exist, and its implementation would hardly find practical application. What is sharding, if even the basic term "scalability" has not received a single generally accepted definition. The blockchain throughput is extremely low: Bitcoin can conduct 3-4 transactions per second, Etherium - up to 15. For comparison, the Visa and Mastercard payment systems can process about two thousand transactions per second. Scalability issues reduce write performance, read performance, and make it difficult to implement blockchains such as Bitcoin in the Internet of Things due to storage scalability issues. This problem is directly related to the issues of sustainable development of the blockchain and requires a comprehensive solution. To organize sustainable development and coordinate consensus algorithms at various levels. It is necessary to imagine the logical structure of the files that we are going to process and their properties: the possibility of concatenation, pausing data for updating, the existence of immutable data types.

2. Results

Despite existing challenges such as node management and reliability, sharding allows for significant increases in throughput due to parallel transaction processing. Analysis of various sharding schemes such as Elastico and OmniLedger, as well as alternative architectures such as directed acyclic graphs (DAG), demonstrates the potential to improve scalability, but requires further research to ensure decentralization and security. In this paper, we highlight the need for a comprehensive approach including dynamic sharding, multi-level consensus, and coordination between sharders to improve network performance and reliability. The proposed conceptual model, including sharding of transactions, states, and networks, contributes to greater scalability and efficiency, which is especially important for high-load applications. Thus, in this paper, we focus on the need to develop innovative protocols and architectures for the sustainable development of blockchain technologies, which is especially relevant in the context of structured financial information and sustainable development.

2.1. The concept of Blockchain Scalability

The concept of "scalability" in the context of blockchain is still not well defined. At present, approaches to defining scalability can be divided into two main categories.

The first category of researchers considers scalability as a complex term that includes such indicators as throughput, latency, the amount of data required for storage, and other system parameters. For example, in the works "Solutions to Scalability of Blockchain: A Survey" and "A systematic review of blockchain scalability: Issues, solutions, analysis and future research" scalability is equated with system metrics such as throughput, storage costs, and latency in reading and writing data. In the study "Blockchain challenges and opportunities: a survey" insufficient scalability is defined by low throughput, high data storage costs, and long latencies in processing transactions and network requests. The second category treats scalability as a fundamental property of a blockchain, similar in importance to such characteristics as decentralization and immutability. In the article "Scalable blockchains — A systematic review", scalability is interpreted based on established concepts in the database field. However, this definition only considers aspects such as throughput and storage costs, which refers to vertical scalability, without taking into account horizontal scalability, which should also be taken into account when measuring these metrics.

The founder of Ethereum described scalability as the ability of a blockchain to handle transaction processing that goes beyond the performance of a single node. In the studies "On The Scalability of Blockchain Systems" and "Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability", blockchain scalability is associated with an increase in system performance with an increase in the number of participants (nodes) without degrading the overall performance level.

In addition, based on the knowledge gained from the database field, we further classify scalability into two types. A blockchain is horizontally scalable if its system metrics improve with an increase in the number of participating nodes. Otherwise, if the blockchain's system metrics can be

 doi:10.20944/preprints202410.1078.v1

improved by increasing the nodes' computing power and storage capacity, or by adjusting parameters such as block size or block interval, then the blockchain is vertically scalable.

Modern information technologies are faced with datasets whose size objectively exceeds (due to technical and computing limitations of the computer) the permissible memory sizes. There is not enough computing power, but the need to analyze these data arrays remains. To solve this contradiction, partitioning was developed - a method of dividing large databases into many small sections (partitions) stored on the same database instance (server). The principles of dividing into sections can be arbitrary, depending on the nature and specifics of the data - by popularity, by date, by geographic location, by price segment.

Sharding as an approach that involves dividing a database into independent segments (shards), each of which is managed by a separate database instance, became a logical consequence of partitioning, which could not ensure the independent existence of the created partitions: if one server fails, all data stored on it will be unavailable. Sharding solved not only this issue, but also the problem of limited memory: using multiple servers provides increased throughput, and we also get the opportunity to use maximum capacity due to the parallel execution of tasks by several processors at once.

There are three fundamental approaches to sharding, distinguished based on the principles of shard formation.

The range-based approach is based on dividing the database orderings into ranges, from which shards are directly formed. For example, let's imagine a database that stores information about the population, GDP, climate, and much more by country. We can sort the records in the database by ascending or descending GDP, identify groups of countries with high, medium and low levels of gross product and form shards on this basis. Or we can split the records into shards depending on the continent on which a particular country is located or based on who predominates in the population - men or women. However, this approach does not always provide an optimal and even distribution of information between shards. It often happens that some shards are overcrowded (hot shards), while others are almost empty. This is explained by the fact that the processed data can be almost identical, their division into ranges is meaningless.

The key-based approach assumes a giant hash function for constructing shards. Let's imagine an arbitrary database, the records in which are numbered with numeric identification numbers ID. Now let's go through the entire database in a loop, applying the hash function to each ID, the result of which will be some value. We will independently determine the required number of shards and find the remainder from dividing each obtained hash function value by the selected number of shards, this remainder shows which shard a particular entry in the database belongs to. The advantages of this approach are the reproducibility of the hash function and a more uniform distribution of data between shards due to configurable encryption parameters.

The directory-based approach is based on creating an intermediate additional table (database), with pre-assigned zones for each shard identifier. Let's take some dataset, see which zone a particular entry is in and correlate this with which shard is responsible for it.

It should be understood that sharding, in addition to all its advantages, is not without its disadvantages. At the moment, there are still no unified algorithms that would track the specifics of processing transactions of each node and could independently optimize the process of storing and transmitting information. There is still no trust mechanism between nodes due to their characteristics, it is replaced by a less effective independent consensus, and the termination of a node requires additional confirmation, which can take a significant amount of time (for Etherium - up to 30-45 minutes). The efficiency of sharding also depends significantly on the consensus algorithm used: the best result is achieved for the PoS (proof of stake) algorithm. In addition, sharding requires the use of replications for each shard, because if one node fails, access to its contents is lost, which is unacceptable. In order to cover these risks, at least 3 replicas are created for each shard. This creates additional costs for data storage. Sharding reduces performance: when accessing a database, the user does not always know which shard stores the necessary information, and it becomes necessary to access all shards in search of the necessary data, which slows down the network. The problem of

 doi:10.20944/preprints202410.1078.v1

determining the shard containing the necessary information can be solved using routing, which allows you to understand which shard to go to for delete, select, insert

Client routing is characterized by the fact that the client knows about the use of sharding in a distributed database. This type of routing allows you to get rid of unnecessary nodes, the client accesses directly those shards that it needs. On the other hand, we are faced with the complication of the client code due to additional logic in the client, in addition, there are difficulties with updating hosts: when adding or deleting nodes, clients must be notified.

Proxy routing involves creating a proxy between the client and the distributed database. On the one hand, this is convenient: the application itself does not know sharding, and it is not required to. On the other hand, this type of routing requires an additional network node, and the proxy itself is a single point of failure and if it fails, access to the database is lost. Moreover, there is a loss of functionality: the proxy is not able to perform complex queries (select, join, group by), unlike another type of routing - coordinator routing. Coordinator is also able to cache certain information within itself. The disadvantages of proxy routing include a significant network load and infrastructure complexity during implementation.

Sharding in blockchain involves dividing the blockchain network into smaller segments or committees to improve scalability and works with a single logical set of data. In blockchain, sharding is used mainly at the transaction level, when sending a message from one node to another. Splitting messages into chunks and processing them in parallel across different nodes on the blockchain enables faster transaction processing.

There are three types of sharding:

- Network sharding - Network sharding reduces network bandwidth requirements by assigning nodes only a portion of transactions.
- Transaction sharding - Transaction sharding divides transactions into sets so that different committees can reach consensus within a shard.
- State sharding - State sharding handles transactions between shards and is necessary for complex applications such as NFTs and smart contracts
- Sharding is aimed at reducing the load on individual nodes, increasing performance and ensuring parallel processing of actions in several shards.

An important element for classifying sharding technologies in terms of scalability is the analysis and classification of the types of data being processed. The use of immutable types facilitates the creation of copies and the sharding process, as well as the use of restrictions that allow, for example, closing a database, copying it, making changes and opening it again. This is important because the main task that we set for ourselves is fast decentralized database processing

There are 4 sharding patterns:

1) transaction sharding and state sharding
2) network sharding and transaction sharding
3) network sharding and state sharding
4) simultaneous adoption of three types of sharding.

 doi:10.20944/preprints202410.1078.v1

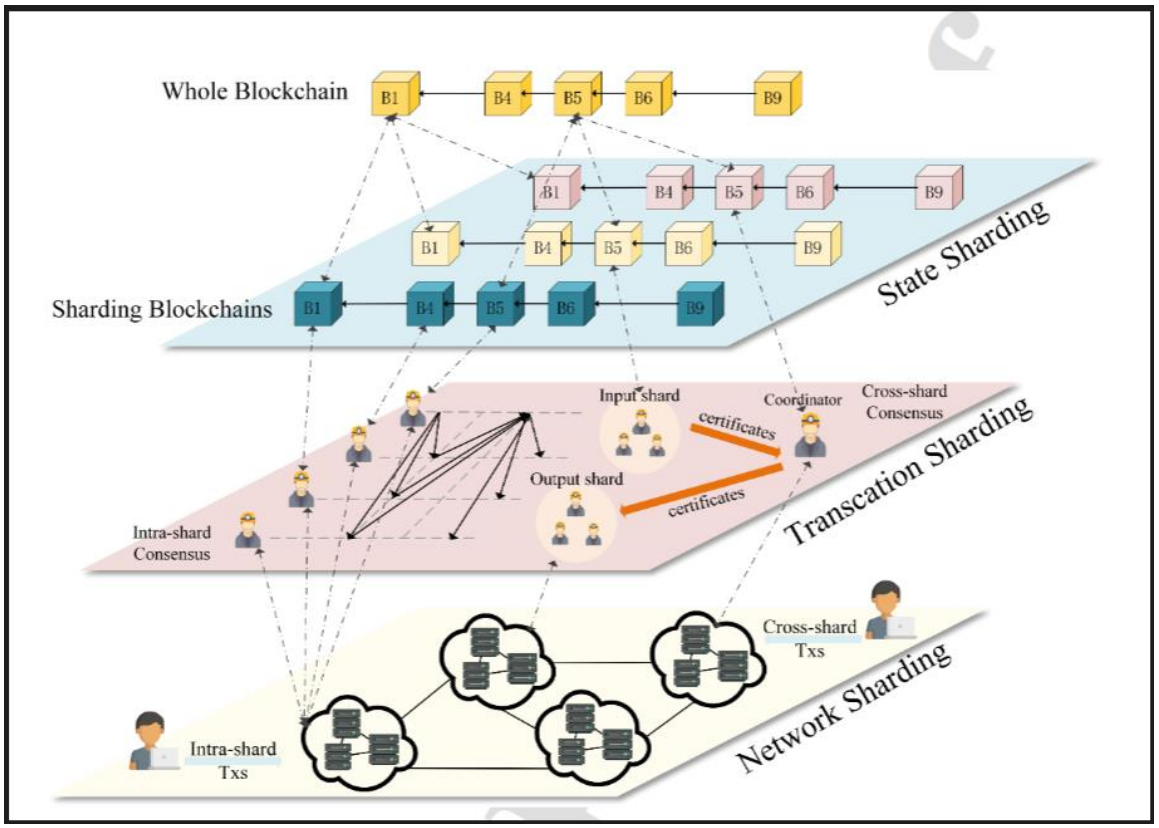

Figure 1. Sharding patterns. Source: "A survey of state-of-the-art sharding blockchains: Models, components and attack surfaces".

2.2. Elastico

Elastico is an innovative blockchain protocol that introduces the concept of permissionless sharding. In this protocol, time is organized into different epochs, at the beginning of each of which nodes run a Proof of Work (PoW) algorithm to generate unique verifiable identifiers. Based on the last s-bits of the identifier, each node is assigned to a specific committee. The main innovation of Elastico is the creation of two types of committees: a director committee and a final committee. The director committee coordinates and forms subsequent committees, thereby ensuring the organization and distribution of tasks. After reaching consensus on the current set of transactions, the director committee passes this set to the final committee. The final committee, in turn, aggregates all legitimate transactions and distributes them for storage among all nodes in the network. Elastico represents the first significant step in applying sharding technologies to blockchain systems. Although it demonstrates a certain level of scalability, the system faces several key challenges. Firstly, Elastico does not support cross-shard transaction processing, which can lead to potential locking of funds and hinder cross-shard communication. Secondly, the need to store a global ledger for each node can significantly reduce its performance, since each node must manage extensive and constantly updated information. Thus, despite the introduction of new sharding technology, Elastico has significant limitations that require further improvement and optimization.

2.3. OmniLedger

OmniLedger was developed as an improved version of Elastico, with the aim of optimizing blockchain data management and increasing the efficiency of transaction processing. Unlike Elastico, which focuses on using permissionless sharding with fixed epochs, OmniLedger introduces a more flexible and adaptive approach. One of the key features of OmniLedger is the use of ByzCoin's sliding window mechanism to determine qualified nodes for the next epoch. This mechanism helps to dynamically determine and update the composition of nodes, providing more flexible network

 doi:10.20944/preprints202410.1078.v1

management and improving its security. The most significant innovation of OmniLedger is the Atomix protocol, which is a client-managed two-phase protocol for processing cross-shard transactions. Atomix effectively guarantees transaction atomicity, which allows transactions to be executed correctly and consistently across different shards. This feature is critical to ensuring data integrity and consistency in distributed systems where transactions may affect multiple shards. Atomix improves system performance and reliability by reducing the likelihood of errors and conflicts that may occur when processing cross-shard transactions. As such, OmniLedger represents a significant improvement over Elastico, providing more efficient and reliable transaction management on a blockchain network.

2.4. Pyramid

Pyramid introduced the concept of multi-layer sharding, expanding on the existing two-layer architecture. In Pyramid, nodes establish their legitimate identities through a Proof of Work (PoW) mechanism and are distributed across different types of shards based on these identities. One such type is the i-shard, which functions similarly to traditional shards and processes and stores intra-shard transactions using the Practical Byzantine Fault Tolerance (PBFT) consensus protocol.

Another type is the b-shard, or bridge shard, which connects multiple i-shards. Nodes in a b-shard represent a kind of union of multiple shards and store all the relevant state of the i-shards. This allows b-shard nodes to independently verify the validity of transactions passing between different shards. To process cross-shard transactions that no b-shard can handle, Pyramid uses a transaction splitting method inspired by Omnileger. This method involves splitting transactions across b-shard connection ranges and then processing them within the corresponding b-shards. While using overlapping shards can significantly improve the efficiency of cross-shard transaction processing, it also introduces two major problems. First, nodes in a b-shard require more computing, networking, and resource capacity to provide adequate quality of service. Pyramid, on the other hand, divides nodes into shards based on identifiers, which can be unpredictable, which does not guarantee that nodes in a b-shard will have sufficient resources to complete their tasks. Second, the configurations of i-shards and b-shards must be fixed and specified in the genesis block before the system goes live. Such a static shard configuration can significantly reduce the security of the system, since it does not allow for dynamic adaptation to changes in the network. Experiments have shown that Pyramid can only withstand 16% of malicious nodes, which is significantly lower than other schemes that use dynamic sharding and reconfiguration, which provide greater resilience to attackers.

2.5. Repchain

RepChain is a sharding scheme that emphasizes the differences between nodes in the network in terms of their activity and role in consensus. Unlike many current sharding systems, where nodes can be divided into functional roles — active, participating in consensus, and inactive, not actively participating — RepChain seeks to balance this imbalance to minimize vulnerabilities in the network. Nodes with high activity and significant contributions to consensus have a higher reputation, which serves as an incentive for honest behavior and incentivizes their active participation.

RepChain's reputation scoring scheme is based on a weighting of each node's contribution to the consensus process. At the beginning of each epoch, nodes are sorted by their reputation values and then assigned to a shard of minimum size based on their ranks. This approach ensures a more even distribution of both the number and quality of nodes in each shard, which helps improve the overall reliability of the network. Nodes with the highest reputation are appointed as shard leaders, where they are responsible for intra-shard consensus and cross-shard transaction processing. This system creates an additional incentive for nodes to strive for a higher reputation, as it directly affects their capabilities within the network. RepChain's shard consensus is innovative due to the use of two data chains: the transaction chain and the reputation chain. To achieve consensus on transactions within shards, the Raft protocol is used, which ensures high throughput and the speed of TB block generation. However, Raft alone is not able to effectively cope with Byzantine nodes, so TB blocks are considered only candidates until they are finally confirmed. For final verification, the reputation

 doi:10.20944/preprints202410.1078.v1

chain is used, which applies the CSBFT consensus algorithm, which ensures an average RB block generation speed. RB blocks contain hashes of several candidate TB blocks, which allows for final confirmation of transactions after consensus on RB blocks is reached. This two-layer structure provides the necessary balance between high throughput and consensus reliability. However, the reputation system in RepChain faces certain challenges related to the attack capabilities of an adversary. If an adversary is able to control a fixed number of nodes and imitate their behavior until one of them becomes the leader, this can lead to an attack on the system. Also, a reputation-based system may be vulnerable to slow-adapting attacks, since it is less flexible in adapting to changes in the behavior of nodes. These aspects highlight the need for careful reputation management and the development of defense strategies against complex attack scenarios.

2.6. SSchain

SSchain implements a two-tier architecture where the first tier is a blockchain root network similar to the original Bitcoin network with a full transaction ledger stored on all nodes. The second tier consists of a network of shards where each shard processes transactions using a Proof of Work (PoW) consensus mechanism. In this structure, the root blockchain performs additional validation of blocks created by shard nodes to prevent double-spend attacks. The root block in SSchain includes transactions from all new blocks generated in shards. Since the root chain and shard networks operate asynchronously, the root block can contain multiple blocks from different shards, resulting in a Directed Acyclic Graph (DAG) structure at the two-tier network level. This DAG structure helps maintain data integrity and prevent double-spends, ensuring high security for the sharding network. SSchain's two-tier architecture does not require random distribution of nodes to ensure shard security, unlike many other systems. Nodes are free to join both shards and the root chain. The root chain ensures the security of the network, which allows SSchain to implement an incentive mechanism that directs the computing resources of the majority of nodes to support the root chain. This helps maintain a high level of consensus security and ensures an adequate number of nodes to simultaneously process transactions in each shard. To improve transaction efficiency, SSchain encourages users to prefer intra-shard transactions, as they are confirmed faster and have lower fees. If a transaction has multiple input addresses and does not require atomicity, the wallet automatically splits it into multiple transactions within a shard. Transactions that cannot be split and require cross-shard processing are forwarded to the root chain. The root chain stores the full transaction ledger and can directly verify and process inter-shard transactions, ensuring their correctness and integrity. This hybrid architecture of SSchain aims to optimize the throughput and security of the system by combining the benefits of high processing speed within shards with reliable transaction verification at the root chain level.

2.7. Brokerchain

Brokerchain is the first to address the hot shard problem that occurs when sharding random states. The hot shard problem occurs when one of the shards containing an account initiating a large number of transactions becomes overloaded, exceeding its maximum computing power. This leads to delays and blocking of transactions both within a given shard and across shards. To mitigate this problem, Brokerchain implements a load balancing and cross-shard transaction pruning strategy. The core component of the system, the p-shard, is responsible for transaction distribution. It continuously collects transactions generated in each shard in the current epoch and applies the Metis algorithm to distribute these transactions across different shards, thereby ensuring a more even distribution of the workload. This helps prevent individual shards from becoming overloaded and improves the overall performance of the system. Additionally, Brokerchain allows a single account to exist across multiple shards, where the overall account status is represented as the sum of all state shards in each shard. This allows for flexibility in managing account state and helps reduce the number of cross-shard transactions. To achieve this, Brokerchain extends Ethereum's Merkle Patricia Trie (MPT) to a Merkle Storage State Tree (MSST). The leaf nodes of the MSST still contain the state of each account, but it also adds a storage map field that indicates the shards in which the account state resides. This map

 doi:10.20944/preprints202410.1078.v1

is represented as a vector, where each element indicates whether a piece of the account state resides in the corresponding shard. In this way, accounts that exist in multiple shards can serve as "brokers" for processing cross-shard transactions. Cross-shard transaction processing in Brokerchain is accomplished using a two-phase lock (2PL) protocol. Brokers act on both source and target shards, converting cross-shard transactions into two intra-shard transactions, improving consistency and reducing the need for frequent cross-shard communication. Brokerchain is a fully sharded solution, storing only transactions and associated account states. However, to ensure account state consistency across shards, each shard must store a map of all account storage. However, the actual balances, code, and other states are not stored in each shard, reducing data volumes and increasing system efficiency.

2.8. Meepo

Meepo is a protocol that focuses on sharding consortium blockchains. Sharding does not occur between Meepo participants, but within each participant's cluster. Nodes in each cluster only process the transactions that are assigned to them. However, transactions in Meepo are broadcast to all participants. This means that each participant owns the full state of the blockchain. However, each node in the cluster only stores a portion of the full state. Nodes of each participant receive transactions on the same P2P network, and network sharding is not used, so Meepo is a partial sharding protocol, not a self-proclaimed full sharding protocol. Meepo uses multiple runtimes on each node to increase throughput, which in turn brings two significant improvements to the processing of transactions between shards. Firstly, since it is a consortium blockchain, all shards in the same cluster (one network participant) trust each other, which reduces the cost of generating and verifying signatures. Secondly, most servers in the same cluster are on the same local area network (LAN), which allows for a fully synchronized network model. Communication overhead can also be ignored.

Meepo also reserves two unique time slots before generating the next block.

- Cross-epoch - during this time, all cross-calls caused by transactions between shards are combined for batch processing.
- Replay-epoch - during this time, erroneous transactions are removed from blocks and then re-executed. This ensures transaction atomicity.

Essentially, Meepo extends a single node in a traditional blockchain into a master-slave computing cluster, where the master server acts as a shard leader responsible for splitting transactions and delivering transactions between other shards. Slave servers execute and store transactions.

Each transaction contains two new fields:

- Shard flag. It is set by the transaction initiator.
- Stage flag. It is set by the contract developer. Each transaction is divided into several stages. Each stage is processed in different shards, thus converting inter-shard transactions into intra-shard transactions. This approach is also known as distributed transactions. A stage is executed in only one shard, and different stages of several transactions can be executed in parallel in different shards.

2.9. AHL

The AHL protocol is an approach to full sharding of blockchain networks. At the core of this protocol is the use of a trusted execution environment (TEE), which provides optimizations for the sharding process. In particular, AHL uses the sgx-read-rand function from Intel SGX, which is designed to generate unbiased random numbers. These random numbers serve as the basis for partitioning nodes into different shards, as well as for periodic reconfiguration of these shards. Random distribution of nodes plays a critical role in preventing attackers from bypassing the consensus protocol, as it helps maintain attack resistance. However, it is worth noting that Byzantine nodes, which are capable of performing arbitrary malicious actions, can be downgraded to common nodes that are limited in their capabilities, such as passing information without bias. In this context, weakened nodes achieve consensus fault tolerance within (n-1)/2, where n denotes the total number of nodes in the network. It is important to emphasize that despite these measures, the AHL approach

 doi:10.20944/preprints202410.1078.v1

only provides probabilistic shard security, not absolute security, due to the inherent limitations of sharding and the security practices of such distributed systems.

2.10. Benzene

The Benzene sharding scheme is the first dual-chain architecture, where each shard simultaneously maintains a proposal chain and a voting chain. This dual-chain structure separates the transaction recording process from the consensus process, facilitating the verification of cross-shard communication without affecting the independence of the transaction recording processes within each shard. In Benzene, each node is required to verify every proposal block coming from all shards. To reduce the overhead of cross-shard communication, an SGX-enabled node hosted in each shard first verifies the validity of candidate blocks and provides proofs of the validity. Then, to request votes, only these proofs need to be propagated to all shards, and the candidate block with the most votes is selected. Compared to non-cooperative designs, Benzene's architecture exhibits significantly greater fault tolerance. Attackers would need to control more than half of the shards to change the voting results, making successful attacks more difficult. However, it is worth noting that Benzene uses a traditional two-phase lock (2PL) protocol to process cross-shard transactions. Each shard must wait for at least k subsequent blocks to ensure that transactions are fully committed, resulting in a latency of over 300 seconds for cross-shard transaction confirmations. Additionally, Benzene's current implementation does not provide a clear explanation of how SGX ensures the validity of the loaded state when validating candidate blocks. If the entire state is loaded into SGX, this could result in memory limits for large states being exceeded. If Merkle proofs are provided for each accessed state, this could slow down the transaction verification process due to frequent context switches and the need to encrypt/decrypt data.

2.11. CycLedger

CycLedger is a sharding system that uses reputation scores to incentivize nodes, similar to the mechanism used in RepChain. In this system, nodes' votes on transactions are considered as contributions to the network. Nodes whose votes are closer to the final consensus result receive larger rewards, which incentivizes them to operate honestly. Nodes with more computing resources are able to process more transactions and, accordingly, receive larger rewards, which encourages their honest behavior. However, CycLedger criticizes the reputation-based node assignment method used in RepChain for de-randomizing the system by only appointing nodes with high reputations as leaders. In response, CycLedger proposes an alternative approach based on the use of a random selection function with proof-of-random-function (VRF). In this approach, a referee committee is responsible for managing node identification and block packaging. In each shard, a node with the highest reputation is elected to become the leader and represent the shard in communication with the referee committee, receiving transactions and other information. In addition, CycLedger introduces the concept of an additional partial set of nodes in each shard that monitor the behavior of the current leader. If one of the nodes in the partial set detects a protocol violation by the leader, this node has the opportunity to replace the leader. Thus, the partial set of nodes acts as a candidate for leadership. CycLedger distinguishes four types of nodes: the judge node (or referee committee node), the shard leader, members of the partial set, and other non-core nodes. However, non-core nodes are required to store only transactions associated with them, while referee committee nodes are required to store all committee transactions for subsequent verification and forwarding. As a result, despite the presence of effective governance and reputation mechanisms, CycLedger can be classified as a partial sharding system, since it does not completely separate the recording and consensus processes in relation to all network nodes.

2.12. Changing the Shard System

 doi:10.20944/preprints202410.1078.v1

One of the most promising technologies in blockchain is changing the shard system, transferring information from one shard to another. Rebalancing is used for this purpose. Let's consider the following system:

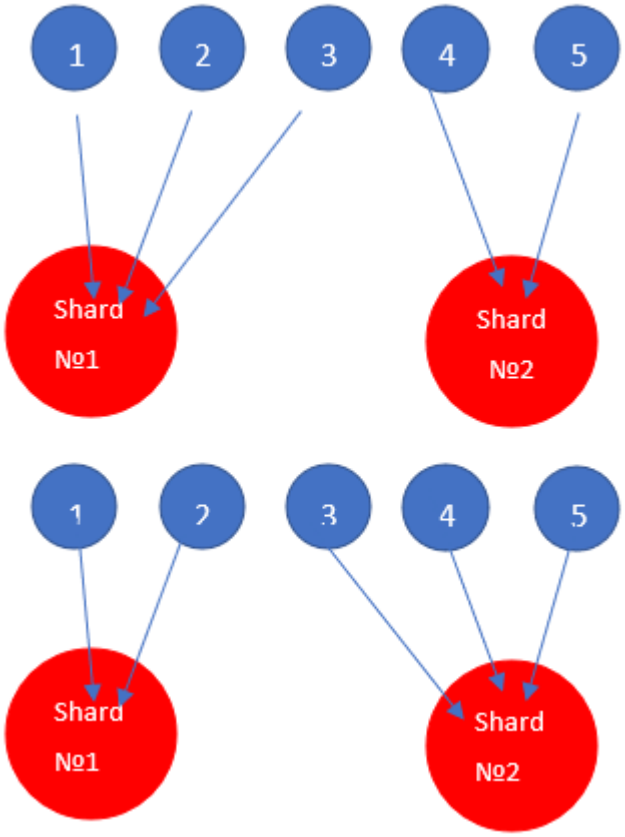

Figure 2. Changing the shard system.

Let's say we need to transfer data from one shard to another. The simplest option looks like this: first, we need to cancel write operations (update or delete writes become unavailable to users for a certain period of time, clients are only given the opportunity to read). At this time (usually at night, during periods of low user activity), we transfer data. However, this scheme is not always convenient, there are situations when we cannot afford the service to be unavailable even for a short period of time, it must be available for writing constantly, then this scheme is not applicable.

Another fundamental case is logs (special text files). They are never changed or deleted, the only applicable operation (maybe with the exception of some exotic cases) is the append operation. Data transfer is carried out by creating an additional shard (target), to which we can write something. Reading can occur simultaneously from two shards - the source (src) and the target (tgt). The most common option is logical replication, which is carried out by creating a replica of the src shard, after synchronizing two shards, the first is cut off. In addition to the above approaches, all sorts of combinations and mixtures are possible.

It happens that the number of shards is insufficient and does not cope with the load, or, on the contrary, is excessive and causes unnecessary expenses, for example, due to an error in choosing a sharding strategy. Resharding was developed to add or remove nodes. Let's assume that we chose a key-based approach to form shards. Let's consider the following schemes:

Let's say there were initially 3 shards. Let's place them on a certain numerical circle, the length of which is normalized by one, and mark the starting point (zero). Now we will pass the database records through the hash function, perform normalization and place the obtained results on the circle. Next, moving clockwise, we will find the closest one in relation to each point and establish a one-to-one correspondence between them. Now, if we want to change the number of shards, for example, to two, we do not have to start the whole process again and redistribute all the records. It will be

 doi:10.20944/preprints202410.1078.v1

enough (in our case) to redistribute the data between the first and second shard. This approach is called the consistent hashing method. There are other methods of resharding, for example, randezvous hashing.

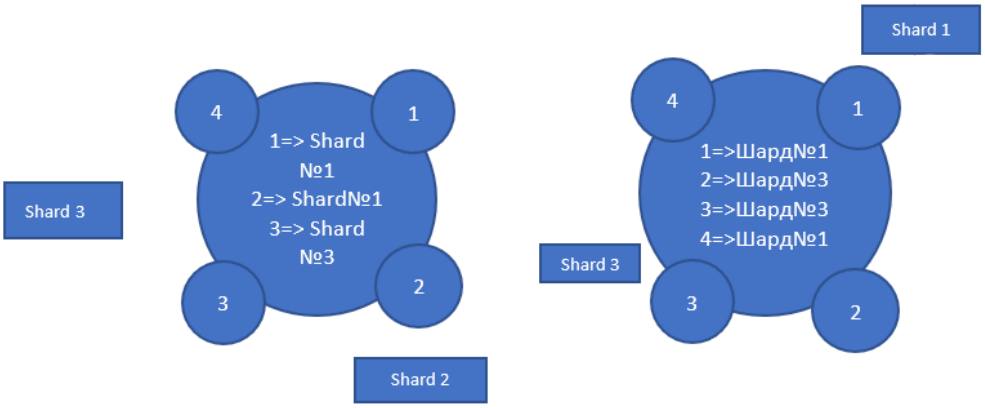

Figure 3. Consistent hashing method.

The growing interest in green investments and the spread of ESG standards have acted as a catalyst for the use of blockchain technologies in working with green investments in general and green bonds in particular.

The implementation of green investments in blockchain networks using sharding requires a thoughtful approach that will take into account both technological and environmental aspects. First of all, it is necessary to create specialized shards that will be dedicated exclusively to the management and tracking of data related to green investments. These shards can contain information on projects financed through green bonds, including data on their environmental performance, as well as compliance with ESG (environmental, social and corporate governance) standards. Such a structure allows for the centralization and organization of data, which significantly facilitates the process of monitoring and managing investments.

Particular attention should be paid to the verification mechanisms within these shards. To ensure maximum transparency and accountability of investments, each shard should include tools for verifying the compliance of projects with established environmental standards and taxonomies. For example, the use of the taxonomy developed by VEB.RF in Russia will help minimize the risks of "green laundering" and ensure the targeted use of funds for truly environmentally significant projects. The integration of such mechanisms into sharding will create a basis for trust on the part of both investors and regulators, thereby ensuring the sustainability of the system.

Further development of the concept includes the adaptation of consensus algorithms to take into account the specific requirements of green investments. An important step may be the development of a multi-level consensus that will allow assessing not only the technical parameters of transactions, but also their environmental aspects. Such an approach may include integration with existing ESG reporting systems, which will help increase transparency and accountability in the decision-making process. At the same time, consensus mechanisms should be flexible enough to adapt to various standards and regulations in force in different jurisdictions, which will allow taking into account both global and local environmental requirements.

Additionally, the possibility of creating inter-shard coordination should be considered, which will ensure the consistency and integrity of data between different shard structures. This can be especially important if there are different shard groups in a blockchain network working on different aspects of green investment, such as assessing the environmental performance of projects, financial management, and ESG compliance. Cross-shard coordination will allow for efficient synchronization of data between these groups, thus providing a more comprehensive and coordinated approach to managing green investment.

Thus, implementing green investment in a sharded blockchain requires a combination of technological innovations with carefully thought-out governance and verification processes that will

 doi:10.20944/preprints202410.1078.v1

ensure not only high network performance and security, but also trust from investors and regulators, thereby promoting sustainable development and supporting environmentally significant projects.

There are other approaches to solving scalability problems. Among them are reducing the size of block data without reducing the number of transactions, directly increasing the block size, using acyclic graphs (DAG), altcoins, parallel data processing, off-chain solutions. Off-chain solutions improve the scalability of the blockchain by executing transactions or tasks outside the main chain.

The set of transactions or completed tasks for a certain period is recorded in the main chain as one transaction. For example, information about money transfers between two users is accumulated, mutual obligations, if any, are repaid and instead of many transactions, only one is performed. This is done in order to unload the main chain, increase its throughput, reduce the load on the storage, and also reduce the transaction fee.

Effective methods of blockchain optimization are the Segwit and MAST protocols. Segwit removes signatures from transaction data and adds them to the metadata along with scripts in the form of a separate structure called Witness. In addition, signatures are now only a quarter of their original size. Signatures take up about 65% of transaction data, so removing them frees up space in the block and allows more transactions to be included in the block (up to 4 times). Therefore, the throughput of transactions per second increases. Segwit also increases the Bitcoin block size from 1 MB to 4 MB. It also solves the quadratic hashing problem and facilitates payment channels such as the Lightning Network, which are other blockchain scaling protocols. Despite the benefits, the throughput improvement in SegWit is limited to 17-23 TPS.

2.13. MAST

Bitcoin allows scripts to be added to transactions. These scripts contribute to the large size of Bitcoin transactions, and some of the scripts cannot be used. An abstract syntax tree (AST) is a way to break software codes into a tree structure in which each block of code is connected to its dependencies until all dependencies are connected. MAST proposed to represent Bitcoin scripts as a Merkle tree of its AST branches. In this way, an unused extra script can be removed from a block. The sender of the coins must provide a Merkle proof of the missing script branch to spend the bitcoins when the Merkle proof returns True. MAST provides a huge reduction in block size on a logarithmic scale.

DAG, or directed acyclic graph, has recently been increasingly considered as an alternative to the existing blockchain architecture. Conceptually, DAG consists of 4 basic elements: nodes, unidirectional edges, nodes without child nodes (sometimes called leaves) and the main ancestor node. DAG, unlike blockchain, which consists of blocks and uses mining to increase the database, stores all data in nodes (vertices), each of which carries information about the transaction. If the blockchain combines new transactions into blocks, then DAG writes them over the old ones. DAG, like blockchain, uses consensus algorithms to confirm transactions. The main advantages of DAG are high throughput and transaction processing speed, no limits on the number of messages sent by users, the rejection of the PoW consensus algorithm and, as a result, ridding the system of mining, and at the same time - a decrease in fees for sending messages (making transactions), which allows attracting a larger number of users, mainly individuals. However, DAG has significant drawbacks: first of all, the security problem.

Directly increasing the block size to increase the number of transactions per second has significant drawbacks: delay in propagation and DOS attacks.

If the algorithm determines the possibility of additional recalculations (the chain is confirmed by the majority of POW POS, this will not be caught, other algorithms are needed) POW POS require improvements in decentralization, scalability and security, in this series and sustainable development, counteracting fraudsters. Blockchain is an established structure, it needs to include program verification and other attributes of sustainable development Limitation on the share of nodes owned by one person (if a person tries to recalculate the algorithm and change the chains,

Conceptual model:

 doi:10.20944/preprints202410.1078.v1

A blockchain model based on three principles of sharding - transaction sharding, state sharding and network sharding - allows for even greater scalability, performance and efficiency. Each type of sharding solves specific problems and interacts with other levels, creating a balanced and sustainable system. We propose the following conceptual model:

2.14. Main Concepts

- **Transaction Sharding:** Dividing and distributing transactions across shard subgroups for parallel processing.
- **State Sharding:** Dividing the global blockchain state across shard subgroups to reduce the load on each individual node.
- **Network Sharding:** Dynamically distributing the node network into groups (shards) to reduce communication overhead and improve load balancing.

Transactions are distributed across shards based on certain criteria, such as sender, receiver, transaction type, or other attributes. Transactions are processed in parallel within each shard, allowing for significant increases in network throughput. Thus, for transactions affecting different shards, a cross-shard coordination mechanism such as atomic cross-shard transactions is used.

The global blockchain state (e.g. balances, assets, contracts) is divided into pieces and distributed across shards. Each shard is responsible for updating its local state as a result of transactions associated with that shard. A global consensus mechanism and inter-shard communication protocols are used to maintain consistency between different parts of the state.

All network nodes are divided into groups (shards) based on geographic location, network characteristics, or random distribution. Network sharding reduces the amount of communication between nodes, since nodes within a single shard communicate more often than with nodes from other shards. Nodes within each shard process transactions and update the state preferentially within their group, which reduces latency and increases processing speed.

Transactions associated with specific parts of the global state are routed to the appropriate shards, which reduces inter-shard dependency. Shard states are updated only in those nodes where this data is actually needed, which reduces the amount of data transferred and increases network speed.

Nodes that are in the same network shard have preferential access to related state data, which improves network performance and minimizes latency. Dynamic network sharding allows nodes and resources to be redistributed depending on the load on different parts of the global state.

Transactions are transferred to the appropriate network shards for processing, which reduces latency and improves throughput. By sharding the network and transactions, the volume of inter-shard communications is reduced, which reduces the load on the network and improves overall performance.

Consensus:

- **Multi-level consensus:** Consensus occurs at several levels - within the transaction shard, within the state shard, and within the network shard. This ensures a high degree of decentralization and security.
- **Inter-shard consensus:** For operations that require interaction between shards, a special mechanism is used to ensure the atomicity and integrity of data.

Network sharding reduces the number of nodes that can be attacked simultaneously, which reduces the likelihood of a successful attack. Attacks that require coordination between multiple shards become much more difficult and require more resources.

The combination of three sharding levels allows the blockchain to scale almost linearly with the addition of new resources (nodes, transactions). Optimization of data transfer and localization of processing within shards significantly increase network performance. Sharding allows for better distribution of computing and network resources, reducing the overhead of maintaining the system.

The blockchain model with sharding of transactions, states, and the network offers an advanced architecture that can cope with the requirements of high-load and decentralized systems. This model

 doi:10.20944/preprints202410.1078.v1

can be used to create blockchains focused on financial applications, IoT, smart contracts, and other areas where high performance and scalability are critical.

The proposed conceptual blockchain model, including sharding of transactions, states, and the network, provides a number of significant advantages for working with structured financial information in transactions:

1. Increased performance through transaction sharding: transaction sharding allows the financial management of transactions to be distributed across multiple independent shards, allowing for parallel execution of transactions. This increases the throughput of the system, allowing for a large number of transactions to be processed per unit of time. This model allows for faster execution of transactions such as payments, fund transfers, or information transfers, increasing system efficiency.

2. Efficient data management through sharding: sharding optimizes the storage and processing of financial information, restoring balance sheet account data, smart contracts, and other important financial objects across different shards. This reduces the amount of data that needs to be stored and processed in each node, reducing computing power requirements and accelerating data access. In financial terms, where high accuracy and efficiency are required, this type of data organization allows for faster and more efficient financial transactions.

3. Reduced network load through network sharding: sharding allows for the division of networks into groups that interact exclusively within their shards. This reduces the amount of network communications required to perform transactions and ensure data consistency, reducing overhead and latency in data transmission. In financial applications, this results in increased transaction speed and reliability, especially for large volumes of transactions that require coordination between nodes.

4. Scalability and adaptability of the system due to the combination of three levels of sharding: the combined use of sharding of transactions, states, and the network allows the system to scale in response to the growth of the number of users and transaction volume. In case of increased load, new shards can be added without degrading the performance of the entire network. This architecture is especially useful for systems that must maintain high speed and stability as the number of clients and transactions grows.

5. High security through decentralization and inter-shard coordination: due to sharding of transactions, states, and the network, the system becomes more resilient to attacks. An attack on one shard cannot completely compromise the entire network, which provides an additional layer of security for financial data. Inter-shard coordination and atomicity of inter-shard transactions ensure that operations affecting multiple shards are executed correctly and consistently. 6. Resource Optimization and Cost Reduction: Network and state sharding allows nodes and network resources to operate more efficiently by processing only the data and transactions that are directly related to their shard. This reduces the cost of computing resources and data storage, making the system more cost-effective. This means lower operational costs while maintaining high performance and reliability.

Thus, a model using sharding of transactions, states, and the network provides a number of advantages that make it especially effective for working with structured financial information. It provides high performance, scalability, security, and resource optimization, which are critical for creating efficient and resilient financial systems.

3. Discussion

4. Materials and Methods

5. Conclusions

The article examines the scalability and security issues in blockchain technologies, highlighting the importance of sharding as a solution to improve network performance and throughput. The focus is on the CAP theorem, which indicates that it is impossible to simultaneously provide consistency, availability, and fragmentation resilience in distributed systems. The paper discusses various

 doi:10.20944/preprints202410.1078.v1

approaches to sharding, including transaction, state, and network sharding, and highlights their advantages and disadvantages. Examples of protocols such as Segwit and MAST demonstrate improved scalability through transaction and block size optimization. Alternative technologies such as DAG, which offer high speed and throughput but face security issues, are also discussed. The conclusion highlights the need for a comprehensive approach to solving scalability and security issues, including the use of innovative protocols and architectures to achieve sustainable development of blockchain technologies. A conceptual model of blockchain is also described, taking into account the full sharding of structured financial information in transactions.

Appreciation: The paper was published with the financial support of the Ministry of Education and Science of the Russian Federation as part of the program of the Moscow Center of Fundamental and Applied Mathematics under the agreement №075-15-2022-284.

References

1. Qiheng Zhou, Huawei Huang, Zibin Zheng, Jing Bian, "Solutions to Scalability of Blockchain: A Survey». IEEE Access, 2020, Volume: 8, 16440 – 16455.
2. Yi Li, Jinsong Wang, Hongwei Zhang, "A survey of state-of-the-art sharding blockchains: Models, components and attack surfaces". Journal of Network and Computer Applications, 2023, Volume: 217, 103686.
3. Abdurrashid Ibrahim Sanka, Ray C.C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research". ". Journal of Network and Computer Applications, 2021, Volume: 195, 103232.
4. Zibin Zheng, Hong-Ning Dai, Shaoan Xie, Xiangping Chen, "Blockchain challenges and opportunities: a survey". International Journal of Web and Grid Services, 2018, Volume: 14(4), 352.
5. Yizhong Liu, Andi Liu, Yuan Lu, Zhuocheng Pan, Yinuo Li, Jianwei Liu, Song Bian, Mauro Conti, Kronos: A Secure and Generic Sharding Blockchain Consensus with Optimized Overheard. Network and Distributed System Security(NDSS), 2024.
6. Muhammad Hassan Nasir, Junaid Arshad, Muhammad Mubashir Khan, Mahawish Fatima, Khaled Salah, Raja Jayaraman, "Scalable blockchains — A systematic review". Future Generation Computer Systems, 2022, Volume: 126, 136-162.
7. Nasrin Sohrabi, Zahir Tari, "On The Scalability of Blockchain Systems". IEEE International Conference on Cloud Engineering(IC2E), 2020, 124-133.
8. M. Kuzlu, M. Pipattanasomporn, L. Gurses and S. Rahman , "Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability". IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14-17 July 2019.
9. Soren Henning, Wilhelm Hasselbring, "A configurable method for benchmarking scalability of cloud-native applications, Scalable blockchains — A systematic review". Empirical Software Engineering, 2022, Volume: 27(6), [143].
10. Huawei Huang, Xiaowen Peng, Jianzhou Zhan, Shenyang Zhang, Yue Lin, Zibin Zheng, BrokerChain: A Cross-Shard Blockchain Protocol for Account/Balance-based State Sharding. IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, London, United Kingdom, 02-05 May 2022.
11. Dorfleitner G., Utz S., Zhang R, The pricing of green bonds: external reviews and the shades of green. Review of Managerial Science, 2021, Volume: 16, 797-834.
12. Ghouma H., Ben-Nasr H., Yan R., Corporate governance and cost of debt financing: Empirical evidence from Canada. The Quarterly Review of Economics and Finance, 2018, Volume: 67, 138–148.
13. Flammer C., Corporate green bonds. Journal of Financial Economics, 2021, Volume: 2 (142), 499–516.
14. Heinkel R., Kraus A., Zechner J., The Effect of Green Investment on Corporate Behavior. The Journal of Financial and Quantitative Analysis, 2001, Volume: 4 (36), 31–449.
15. Sheng Q., Zheng X., Zhong N., Financing for sustainability: Empirical analysis of green bond premium and issuer heterogeneity. Natural Hazards, 2021, Volume: 107, 2641-2651.
16. Ovechkin D.V., Responsible Investments: Divergence of Esg-Ratings. Modern Economy Success, 2021, No. 1.
17. Bezsmertnaya E. R., Issue of "green" bonds as an element of the environmental protection system. Economy. Taxes. Law, 2019, Volume: 5 (12), 61-69.
18. Ovechkin D. V., Responsible investments: the impact of ESG rating on profitability firms and expected returns on the stock market. Scientific Journal of Niu ITM. Series: Economics and Environmental Management, 2021, Volume: 1.
19. Yumanova N. N., Bolgov M., A. Development of the Green Bond Market in Russia. Russian Economic Bulletin, 2021, Volume: 1 (4), 211-228.

 doi:10.20944/preprints202410.1078.v1

20. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. August 21, 2008, 9 pages – Access mode: URL: https://archive.org/details/BitcoinAPeer-to-PeerElectronicCashSystem
21. Castro, M., & Liskov. Practical Byzantine Fault Tolerance. Appears in the Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, USA, February 1999, 14 pages.
22. Buterin, V. Ethereum White Paper, 2014 – Access mode: URL: https://ethereum.org/en/whitepaper/
23. King, S., & Nadal, S. (2012). PPCoin: A peer-to-peer proof-of-stake cryptocurrency. – Access mode: URL: https://archive.org/details/PPCoinPaper
24. Rosenfeld, M. (2014). An analysis of reward systems for sharing Bitcoin mining. – Access mode: URL: https://www.semanticscholar.org/paper/Analysis-of-Bitcoin-Pooled-Mining-Reward-Systems-Rosenfeld/19e5af9721409f13496bb4f1635f98a18c7d7e68