Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Review paper

A Comprehensive Review on Blockchain Scalability

A. Matani, A. Sahafi *, A. Broumandnia

Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.

Article Info

Article History:
Received 30 July 2023
Reviewed 25 September 2023
Revised 05 October 2023
Accepted 05 November 2023

Keywords:
Blockchain
Scalability
Consensus
Sharding
Throughput

*Corresponding Author's Email Address: *sahafi@iau.ac.ir*

Abstract

Background and Objectives: Blockchain technology as a distributed and tamper-proof data ledger is attracting more and more attention from various fields around the world. Due to the continuously growing of the blockchain in both transaction data and the number of nodes joining the network, scalability emerges as a challenging issue.

Methods: In this survey, the existing scalability solutions in the blockchain are discussed under five categories including on-chain scalability, off-chain scalability, scalable consensus mechanisms, DAG-based scalability, and horizontal scalability through sharding. Meanwhile, the novelties they have created on the fundamental layers of the blockchain architecture are investigated.

Results: As a result, the advantages and disadvantages of the discussed mechanisms are pointed out, and a comparison between them in terms of different scalability metrics such as throughput, latency, bandwidth, and storage usage is presented. Therefore, this study provides a comprehensive understanding of the various aspects of blockchain scalability and the available scalability solutions. Finally, the research directions and open issues in each category are argued to motivate further improvement efforts for blockchain scalability in the future.

Conclusion: Scalability allows blockchain system to sustain its performance as it grows up. Lack of scalability has a negative effect on the mass adoption of the blockchain in practical environments. This paper presents a profound analysis of the existing scalability solutions, the issues and challenges they address, and the ones that are not resolved yet. Consequently, it inspires novel ideas for more scalable and efficient blockchains in the future.

Introduction

Blockchain is a distributed ledger that eliminates the need for any third-parties and enables the participating nodes in a peer-to-peer network to agree on every single entry in the data ledger using a consensus mechanism. Thanks to interesting features such as decentralization, immutability, transparency, and trustlessness, blockchain has turned into the most breakthrough technology and has been used in developing the applications of diverse domains. For example, many works take advantage of blockchain technology in finance [1], [2], E-Healthcare [3], [4], Internet of Things [5]-[7], supply chain [8], [9], Electricity management [10], [11], insurance [12] and voting [13], [14].

Numerous works proceed research to address different challenges associated with blockchain systems e.g. security [15], [16], decentralization [17], [18], scalability [19]-[22], query processing [23]-[25], blockchain indexing [26], [27] and so on. Generally, in order to carry out a robust project, it is essential to make a trade-off among three key properties of blockchain including decentralization, security, and scalability. Vitalik Buterins, one of the co-founders of Ethereum [28], claims that the blockchain systems can only have two out of these three properties and refers to it as a scalability

trilemma. It is obvious that security is a vital feature for blockchain and not able to be sacrificed for any two other properties. On the other hand, decentralization is an intrinsic feature of blockchain systems. Therefore, scalability remains a challenging feature that should be handled without compromising security and decentralization. Scalability enables blockchain to manage the growing number of requests effectively and retains its performance over different aspects e.g. throughput, latency, storage, and bandwidth usage, as it expands. Due to the distributed and agreement-based nature of the blockchain, its throughput in terms of Transactions Per Second (TPS) is considerably low in comparison with traditional databases. In addition, to participate in the consensus-making process, the nodes need a huge storage space to maintain a copy of the data ledger and sufficient bandwidth to communicate with other peers during the consensus process. Consequently, the mentioned scalability issues can become bottlenecks, hindering the widespread adoption of blockchain technology. A number of solutions have been proposed in the literature to cope with these issues. This survey reviews some of the top-cited research works addressing scalability issues and groups them into 5 categories: (1) on-chain scalability, (2) off-chain scalability, (3) scalable consensus mechanism, (4) Directed Acrylic Graph (DAG)-based scalability, and (5) horizontal scalability through sharding.

On-chain scalability strategies [29]-[38] are aiming to improve scalability by modifying the core features and elements of the blockchain like block [31], [38] or transaction structure [29], [30]. On the other hand, off-chain strategies [39]-[46] are designed to leave transaction processing outside the blockchain to save storage space and mitigate blockchain workload. For instance, Lightening Network [39] and Raiden Network [40] have adopted this strategy to enhance scalability. Scalable consensus mechanisms [47]-[55] refer to the methods that lead to agreement on a greater set of transactions in a shorter time. The fourth category, namely DAG-based scalability, points to the solutions in which instead of traditional blockchain, an alternative data structure named DAG is used [56]-[62]. Generally speaking, in these methods, the data ledger is modeled as a directed acyclic graph with vertices representing users/accounts and edges representing transactions among them. Hence, the transactions can be processed independently resulting in a significant increase in the throughput of the data ledger. Finally, horizontal scalability through sharding implies solutions that periodically partition blockchain nodes into subsets called "shards" and allow parallel processing of the transactions in shards. Sharding is the most promising approach towards improving scalability, and sharding-based protocols [63]-[72] have achieved a high improvement in throughput and other scalability criteria. In the following, related works are investigated in more detail. The contributions of this paper are as follows:

- First, to provide a background of the blockchain components, a layered architecture of the blockchain along with key components within each layer is discussed.
- Then, existing scalability solutions are organized into a taxonomy and their ways of improving the blockchain scalability besides their advantages and disadvantages are debated.
- In addition, the solutions of each taxonomic category are compared in terms of their key characteristics and scalability improvements including throughput, latency, storage, and bandwidth/ communication overhead.
- Finally, the remaining issues and future research directions for each category of the scalability solutions are individually outlined.

The remainder of this paper is as follows. The next section gives the preliminaries of the blockchain. After that, the existing surveys in blockchain scalability are reviewed and compared with this work. Then, the research methodology followed by this paper is explained. In the following sections, a taxonomy of the blockchain scalability solutions is presented, existing works are surveyed in detail and future research directions and open issues are discussed. Finally, the last section concludes the paper.

Blockchain: Preliminaries

In this section, aiming to achieve a better comprehension of the subsequent explanations, a general architecture of the blockchain along with some fundamentals is described. According to the abstraction layer model suggested by the authors in [73], the blockchain architecture is comprised of five layers: (1) data layer, (2) network layer, (3) consensus layer, (4) execution layer, and (5) application layer. In the following, the functionalities of these layers and key components within each layer are explained.

A. Data layer

The data layer in the blockchain architecture is responsible for data management in blockchain systems. The main focus of this layer is on the data structure, transaction model, and cryptographic mechanisms such as digital signature, Merkle tree and hash function, that ensure the security and integrity of information stored on the blockchain.

I) Types of Data Ledgers

From the perspective of data structure, the data ledgers are divided into two main categories: blockchain (e.g. Bitcoin [74] and Ethereum [28]) and DAG (e.g. IOTA [56] and Nano [57]).

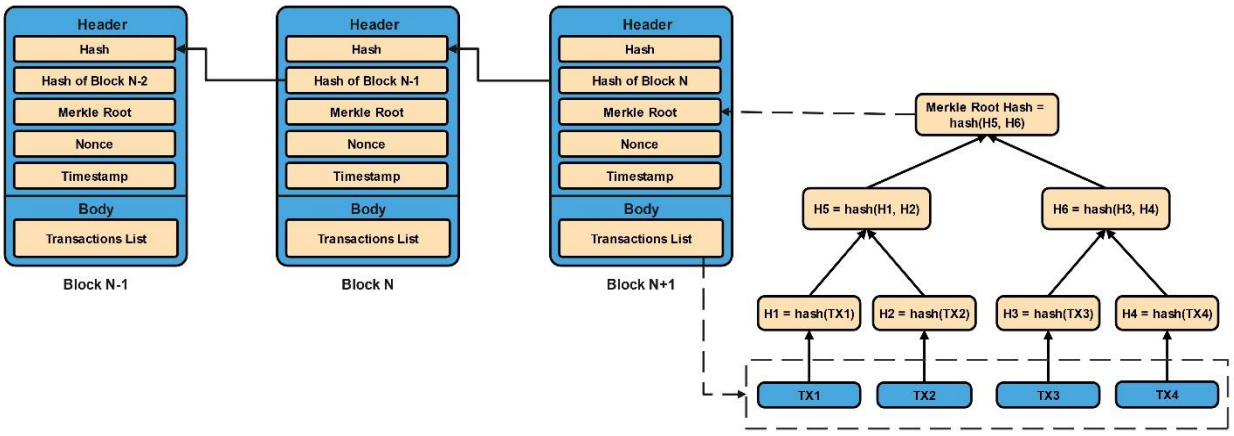

Fig. 1: Structure of blockchain [75].

Blockchain is a back-linked list of blocks chained together in an immutable and chronological order. As Fig. 1 [75] shows, to chain blocks together, each block is linked to the existing blockchain using the hash of the previous block. Each block consists of a set of verified transactions that are grouped together by a miner to be registered on the data ledger.

Opposite to the blockchains, in DAG-based ledgers, there is no chain of blocks and the data ledger seems like a graph. In other words, DAG is a network of individual transactions that are linked together and provide validation for each other. Practically, each new transaction must validate previous transactions and reference them to be registered on the network for validation. Therefore, the transactions that are directly or indirectly referenced by a given number of the transactions can be considered as committed.

Hence, there is no need for miners to mine blocks of transactions, resulting in fast confirmation times of the transactions and subsequently improving throughput and scalability. For example, in Fig. 2 [76], a weight is assigned to each transaction, and a transaction is considered as committed if the cumulative weights of the transactions which confirm it, be equal to or greater than 4 (as a threshold).

Other than blockchain and DAG, there exist other types of data ledgers that have been used in some data ledgers like Codra [77] and Radix [78].

II) Types of Transaction Models

The transaction is the main element for storing and exchanging information on the blockchain. Each transaction causes a blockchain transition from a valid state to another valid state.

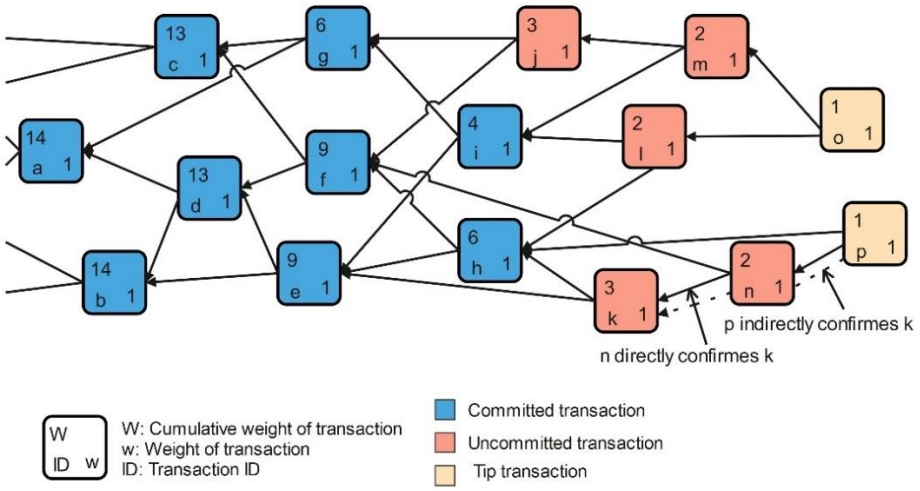

Fig. 2: An example of DAG structure named Tangle [76].

Two popular transaction models used in the blockchains are: Unspent Transactions Output (UTXO) and account-based. In the UTXO model, each transaction spends unspent outputs owned by the sender to create new outputs for a receiver as the new owner. In other words, assets owned by a user are scattered across the data ledger as unspent outputs of the transactions received by that user. The main advantage of the UTXO model is that it facilitates parallel processing of the transactions due to its atomicity and thus provides better infrastructure for scalability solutions. The bad point about the UTXO model is that it is only suitable for the applications in which each output is owned by one person. Moreover, it complicates the development of state-full smart contract-based applications because of its stateless nature.

In contrast, the account-based transaction model, analogous to the traditional banking model, maps each account into a balance. It has to be said that, the accounts' balances are stored in a global state trie that is constantly updated. Each transaction updates the global state as it deducts an amount from the balance of the sender and then adds that amount to the balance of the receiver. In comparison with the UTXO model, the account-based model is simpler and more efficient because transaction validation only needs to check whether the sender account has enough balance or not. In addition, it facilitates the development of smart account-based applications, specially state-full and multi-party ones. Nevertheless, scalability in account-based systems is more challenging than UTXO-based ones. Finally, it is worth mentioning that Bitcoin [74] uses the UTXO transaction model, whereas Ethereum [28] uses the account-based transaction model.

III) Cryptographic Components

In order to keep transactions data secure and immutable, blockchain uses cryptographic mechanisms, namely Merkel tree, hash function, digital signature, and Public Key Infrastructure (PKI) simultaneously. Merkle tree and hash function are used together to provide data integrity, whereas data signature is used to verify the authenticity of the transactions and ensure non-repudiation.

a) Merkle tree and Merkle Patricia Trie

In the blockchain, in order to chain the blocks in a tamper-resistant manner, each block header includes the hash of the previous block. Hence, any modifications in previously published blocks require changing all the subsequent blocks because they include the hash of the modified block.

In addition to the hash of the previous block that guarantees the integrity of the past transactions' history, each block contains a Merkle tree root that provides integrity for the current block's transactions (Reference to Fig. 1 [75]). Merkle tree is a binary tree in which each leaf node contains a hash of a transaction while each non-leaf node contains concatenated hashes of its children. Therefore, the Merkle tree root is a hash value obtained from the hash of all the transactions in the current block and any alternations in transactions will be detected by other nodes in the network because the Merkle root of the altered transactions will not match the one stored in the block header.

Bitcoin [74], in which transactions are the only state, uses the Merkle tree for the above-mentioned purposes. On the other hand, Ethereum [28], in which each node stores a global state consisting of a mapping between accounts and the account state, uses a Merkle Patricia Trie (MPT) that is an implementation of the Modified Merkle Patricia Trie [79]. MPT is a cryptographically authenticated key-value mapping that is used for storing and retrieving the accounts' state, as well as verifying data integrity. In MPT, leaf nodes store key-value states where the value is the account state and the key is its hash, whereas non-leaf nodes store the hash of the next node. Therefore, retrieving an account state needs to traverse MPT downward through the non-leaf nodes each of which stores the key of the next node, until reaching the leaf node storing the value corresponding to the searched key. The MPT allows checking data integrity by computing the Merkle root hash of the trie since if any key-value pair is modified, the Merkle root hash will not match for the entire list of the key-value pairs.

It must be pointed out that both the Merkle tree and MPT allow verifying the inclusion of a state (i.e. key-value state in MPT and transaction in Merkle Tree) without access to the entire blockchain using a method called Simplified Payment Verification (SPV).

After all, it is evident that the hash function is a fundamental component of blockchain technology. Most blockchains use the SHA-256 hashing algorithm, however, other hashing algorithms such as SHA-3 and Ripemd160 have been used by several blockchains.

b) Public Key Infrastructure and Digital Signature

In PKI technology, each user owns a pair of keys: a public key and a private key, which are used for authenticating users and protecting sensitive data. The public key is distributed on the network and is known to other nodes while the private key must be kept secret to never be known by any other nodes except its owner. PKI has algorithms that enable participating nodes in a network to encrypt, decrypt, sign and verify messages using their pairs of keys.

In PKI, if a message is encrypted with one key, it can only be decrypted with the second key. Therefore, if a message is encrypted with the public key of the receiver, it can only be decrypted with the private key of the receiver. In this case, the encrypted message is protected

from eavesdropping by malicious users since the receiver account is the only one that knows its private key and can decrypt the message. On the other side, if a message is encrypted with the private key of the sender, it can only be decrypted with the public key of the sender. In this case, the encrypted message is authenticated in terms of its source because the sender account is the only one who knows its private key and can encrypt that message. Hence, if a message is encrypted by the private key of the sender, the entire encrypted message serves as a digital signature since it ensures a receiver that the message has been encrypted by a claimed sender.

The digital signature is a primary usage of PKI technology in the blockchain. In fact, a digital signature is a mathematical function used to present the authenticity of the transactions and ensure non-repudiation and data integrity. Therefore, each transaction is signed by the private key of the sender to be authenticated by other participating nodes in the blockchain. Elliptic Curve Digital Signature Algorithm (ECDSA) is the most widely used data signature algorithm that has been used by Bitcoin [74] and numerous blockchain applications, although some blockchains use different digital signatures such as Edwards-curve Digital Signature Algorithm (EdDSA) [80], Borromean Ring Signature (BRS) [81], and One-Time ring Signature (OTS) [82].

In addition, a number of works [63], [66] use PKI combined with the Proof of Work (PoW) to establish identities for users securely and unpredictably.

B. Network Layer

Blockchain operates on a peer-to-peer (P2P) network that allows nodes to join the network and communicate with each other in a trustless way. The network layer is characterized by P2P network topology, peer discovery, identity management, block and transaction propagation, and takes care of privacy, anonymity, communication cost, security and attack resiliency.

I) Types of Nodes

There are two types of nodes in the network layer: lightweight nodes and full nodes. Lightweight nodes only store block headers and verify transactions by the SPV method, which allows users to verify the inclusion of a transaction in a block using a Merkle path and referencing to a trusted full node, whereas, full nodes store a complete and up-to-date copy of the blockchain and verify transactions autonomously without any external references. A full node is more reliable and safer than a lightweight node, however, it needs more storage space, bandwidth and computing power than a lightweight node.

II) Types of Blockchains

From the perspective of accessibility, blockchains are classified into two primary types: public and private. A public blockchain is open to the public so everyone can join the network. On the other hand, a private blockchain is closed to the public and each user requires to be authorized for joining the network.

Additionally, from the perspective of permission, blockchains are divided into two types: permission-less and permissioned. In a permission-less blockchain, each participating user can read, write or validate transactions without specific permission, whereas in a permissioned blockchain, authorized users need to obtain permission to read, write or validate transactions.

Finally, based on accessibility and permission, blockchains can be classified into four groups:

- Public permission-less (e.g., Bitcoin [74], Ethereum [28], Litecoin [83])
- Public permissioned (e.g., Ripple [84], EOS [85], Sovrin [86])
- Private permission-less (e.g., LTO [87], Holochain [88], Monet [89])
- Private permissioned (e.g., Hyperledger [90], Corda [77]).

C. Consensus Layer

The Consensus layer is a key aspect of the blockchain because in order to ensure consistency between the copies of the data ledger spread across the P2P network, the full nodes need to achieve a consensus on any updates to the data ledger. Essentially, the consensus process has an important role in many aspects of the system performance, such as scalability, integrity, and security. Consensus algorithms could be grouped into three following types: (1) proof-based, (2) vote-based, and (3) DAG-based.

In proof-based consensus algorithms, nodes compete to obtain the right to append the next block to the chain and the node that proves sufficient proof of qualification will win the competition. Proof of Work (PoW) and Proof of Stack (PoS) are the most popular proof-based consensus algorithms. For example, Fig. 3 [91] represents the flowchart of the PoW consensus process where the nodes need to prove their computational effort to add (mine) a new block to the network. To do so, a miner generates a random number (referred as to Nonce) and combines it with block data so that the hash value of the output data will be less than or equal to the current target of the network. The Proof-based consensus algorithms are appropriate for public blockchains since they provide high security in a trust-less system and also can easily scale in the number of users. Despite these advantages, this type of consensus algorithm reveals low transaction output and also the majority of them (e.g. PoW) are computation intensive and prone to the 51% attack occurring when a single node or a group of the nodes obtains control of more than 50% of the blockchain's

mining power.

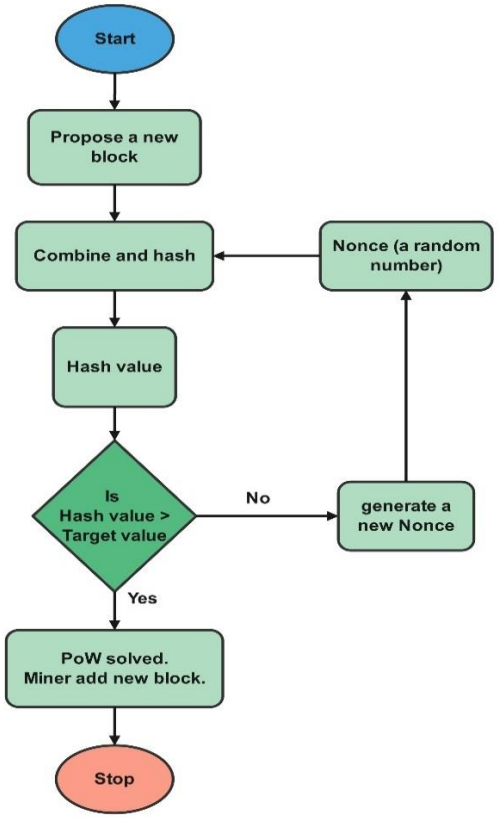

Fig. 3: Proof of Work (PoW) flowchart [91].

Oppositely, in the vote-based consensus algorithms, a leader is first elected to propose the next block. Then, the elected leader announces the next block to the other nodes having voting right. Afterward, each node participating in the voting process validates the proposed block and multicast necessary messages to the other nodes. Finally, if a given number of the nodes agree on a new block, it can be appended to the chain. Byzantine Fault Tolerance (BFT)-based algorithms such as Practical Byzantine Fault Tolerance (PBFT) [49] and Raft [92] are some popular examples of this type of consensus algorithm. The vote-based consensus algorithms have better transaction output and lower latency than the proof-based ones, although they are communication intensive and difficult to scale especially in large-scale environments. Hence, the vote-based consensus algorithms only work well on private and permissioned blockchains.

On the other hand, DAG-based consensus algorithms are used in the data ledgers that adopt DAG as their data structure, where a new transaction requires to validate the previous transactions in order to be processed by other transactions. In other words, in the DAG-based consensus algorithms, transactions provide validation for each other and can be processed in parallel, leading to fast transaction confirmation times. Fig. 2 [76] presented in Section "*Types of data ledgers*" depicts an illustration of a DAG-based data ledger where transactions are validated by each other. The DAG-based consensus algorithms have a comparative advantage in performance, scalability and simplicity, although their security can be compromised by malicious users who validate their transactions.

D. Execution Layer

The execution layer offers a runtime environment enabling nodes to participate in the network and interact with each other. A runtime environment is composed of Virtual Machines (VMs), compilers and containers that are installed on the computers and allow them to operate as a blockchain node. VMs contain APIs and services that enable nodes to execute and validate transactions, organize them into blocks and then share blocks with other peers.

Ethereum blockchain has developed its own virtual machine called Ethereum Virtual Machine (EVM). Ethereum nodes run EVM to execute smart contract code. A smart contract is a computer program that is executed automatically by nodes under predefined conditions. The smart contract helps transactions to be executed in a secure, transparent and conflict-free way. Smart contracts are written in a high-level language named Solidity [93]. Therefore, in order to run on an executing machine, smart contracts first need to be compiled into bytecode by the Solidity compiler, then these bytecodes are executed by EVM and deployed on the blockchain.

E. Application Layer

The application layer provides an interface for blockchain users to easily interact with the network, see results, share information and so on. In other words, the application layer is the first layer used by users to communicate over the blockchain network. Therefore, the usability and efficiency of blockchain applications greatly depend on the flexibility, speed and agility of this layer. Cryptocurrency providing a gateway for exchanging digital currency is the most popular example of the application layer. other examples are Decentralized Applications (DApps) developed in different domains and industries.

Existing Surveys

This section summarizes the existing surveys on blockchain scalability and outlines their contributions.

Hafid et al. [94] surveyed blockchain scalability under two categories: first-layer and second-layer. To enhance scalability, first-layer solutions modify the core features of the blockchain, whereas second-layer solutions are implemented outside of the blockchain and built on top of it.

In [94], sharding-based solutions along with other solutions including DAG-based and bigger block solutions are placed in the first-layer solutions, although the focus

is more on the sharding-based solutions. Hafid et al. [94] presented a taxonomy of sharding-based solutions based on committee formation and intra-committee consensus. However, Hafid et al. [94] reviewed a comprehensive range of the scalability solutions, they did not discuss the solutions in enough detail (specially sharding-based solutions).

Another disadvantage is that the discussed future works do not cover all the solution types and are only regarding sharding-based solutions.

Zhou et al. [95] presented a review of the blockchain scalability solutions and classified them into 3 layers: (1) layer 1 solutions, which are interrelated to the block data, consensus strategies, sharding, and DAG-based data ledgers, (2) layer 2 solutions that are associated with non on-chain techniques and include payment channel, side-chain, cross-chain, and off-chain computation mechanisms, and (3) layer 0 solutions whose main concern is to optimize data propagation in the blockchain. Despite the careful subdivision of the solutions in these layers, some key related works are not investigated in detail and their main contributions are not well defined. The main weakness of the survey [95] is that it lacks a comprehensive comparison of the discussed works and only a few methods are compared in terms of the throughput and confirmation time.

Differently from the aforementioned surveys [94], [95] the survey conducted in this paper provides a more detailed taxonomy of the blockchain scalability solutions including 5 categories: (1) on-chain scalability, (2) Off-chain scalability, (3) scalable consensus mechanisms, (4) DAG-based scalability, and (5) Horizontal scalability through sharding.

Exploiting such accurate taxonomy helps to discriminate and compare the key features of the various solutions precisely. Therefore, in this paper, a comprehensive comparison is presented separately for each category. In addition, the future work for each category is highlighted individually. Another advantage of this survey is that it compares the discussed solutions in terms of various scalability measurements including throughput, latency, storage usage, and communication overhead.

Nasir et al. [96], presented a systematic survey in which they define two dimensions for blockchain scalability: horizontal scalability and vertical scalability. Horizontal scalability refers to scaling blockchain by adding more nodes and clients, whereas vertical scalability refers to boosting the capabilities of the participating nodes such as processing power, storage capacity, memory, and efficient strategy. Vertical scalability is further broken down into several sub-dimensions including throughput, latency, block generation rate, and storage (chain size and block size). Nasir et al. [96] categorized scalability solutions into 5 groups including: (1) on-chain solutions, (2) off-chain solutions, (3) hardware-assisted approaches, (4) parallel mining/ processing, and (5) Redesigning blockchain, although they did not go deep into the solutions of each category. In addition, scalable consensus mechanisms have not been investigated in the survey [96].

Yu et al. [97] provided a survey focusing on the sharding solutions. They presented a comprehensive comparison of the key features of the sharding-based solutions and also conducted a systematic analysis of the scalability metrics such as throughput, latency, storage and communication complexity, although the debated solutions are limited to only a small number of sharding-based solutions.

Wang et al. [98] provided an overview of state-of-the-art DAG-based blockchains and also abstracted a general model to describe them in a theoretical and mathematical form and then identified 6 types of DAG-based blockchain systems.

They evaluated and compared the studied systems from the perspectives of their structure, consensus mechanism, security, and performance (in terms of scalability, throughput, and latency).

Oyinloye et al. [99] presented a comprehensive overview of the alternative consensus protocols which have been proposed in recent years, even the lesser-known ones. They evaluated the alternative consensus mechanisms in terms of throughput, scalability, security, energy consumption and block/ transaction finality (including absolute/ immediate finality and probabilistic finality).

Therefore, the main advantage of this survey over tree above-mentioned surveys [97]-[99] is that it covers a comprehensive variety of scalability solutions, instead of focusing only on the sharding-based solutions or DAG-based solutions and alternative consensus protocols. Table 1 presents a summary comparison between this work and the described surveys.

Research Methodology

This survey is accomplished based on four Research Questions (RQ) and is aiming to answer these questions at the different steps of the study. The questions are as follows:

- RQ1: What are the scalability bottlenecks in the blockchain systems?
- RQ2: What metrics are used to measure blockchain scalability?
- RQ3: Which blockchain elements can be manipulated to improve scalability?
- RQ4: What are the open issues and future prospects for the blockchain scalability?

The research methodology of this survey consists of 6 steps that are described below:

Table 1: Comparison Between this work and existing surveys

Reference	year	Publisher	Covered years	Covered Scalability solutions					Evaluation metrics
				On-Chain	Off-Chain	Consensus mechanism	DAG	Shrading	
This Work	--	--	2014-2021	✓	✓	✓	✓	✓	Throughput, Latency, Storage, Bandwidth
Hafid et al. [94]	2020	IEEE	2014-2020	✓	✓	✓	✓	✓	Throughput, Latency
Zhou et al. [95]	2020	IEEE	2014-2019	✓	✓	✓	✓	✓	Throughput, Latency
Nasir et al. [96]	2021	Elsevier	2015-2020	✓	✓	×	✓	✓	Throughput, Latency, Block generation rate, Storage
Yu et al. [97]	2020	IEEE	2016-2019	×	×	×	×	✓	Throughput, Latency, Storage and Communication complexity
Wang et al. [98]	2020	arXiv preprint	--	×	×	×	✓	×	Scalability, Throughput, Latency
Oyinloye et al. [99]	2021	MDPI	2018-2020	×	×	✓	×	×	Throughput, Scalability, Security, Energy consumption, Finality

F. Keywords Generation

To generate the keywords for searching the relevant research works, first, some possible answers were provided for RQ1 and RQ2.

Then, two sets of keywords were extracted from the answers of the RQ1 and RQ2, respectively named K1 and K2.

As can be seen in Table 2, K1 and K2 keyword sets were combined with K, a keyword set including primary keywords such as blockchain, scalability, scalable and scaling, to generate the expressions for searching among electronic databases (considering the synonyms words).

Table 2: Process of generating the keywords and searching expressions

	Set	Keywords
Main Keywords for blockchain scalability	K	blockchain, scalability, scalable, scaling
RQ1: What are the scalability bottlenecks in the blockchain systems?	K1	network size, blockchain size, high communication overhead, storage, block size, consensus (inefficient consensus strategies)
RQ2: what metrics are used to measure blockchain scalability?	K2	throughput, latency, storage usage, bandwidth
Search expressions = ((k1 or k2) and k) where {k ∈ K, k1 ∈ K1, k2 ∈ K2}		

G. Searching of Research Works

In this phase, using the generated search expressions, the research works were searched in electronic databases such as IEEE, Springer, ACM, Elsevier, Google Scholar, Taylor & Francis and so on. At last, 137 research works containing the mentioned keywords were found.

H. Refinement of Research Works

To select more relevant and valuable research works, among the 137 discovered researches, the ones having the below conditions have been excluded from the study:

- The papers not written in English.
- The papers published before the year 2014 (except the highly cited ones).
- Short papers with less than 8 pages (except the highly cited ones).
- The low-citation papers that have been published before the year 2019.
- The review papers.

The output of this phase is 33 research papers.

I. Cross Checking the Selected Research Papers

In this phase, the references of the selected papers in the previous phase were checked out, to ensure not missing the important and valuable researches.

This checkout resulted in finding 8 other papers. Therefore, during this study, totally 41 research works in the field of blockchain scalability have been studied in

detail.

J. Classification and Review

Finally, the selected papers were studied and categorized based on the strategy that they have applied to enhance blockchain scalability.

K. Identification of Future Research Directions

Meanwhile evaluating the research works, the unresolved issues and also some promising directions were identified and have been recommended at the end of this study for future works.

Fig. 4 illustrates a summary of the steps followed by the methodology of this survey in sequential order. Fig. 5 and Fig. 6 also respectively show distribution of the reviewed research works by publisher and publication year.

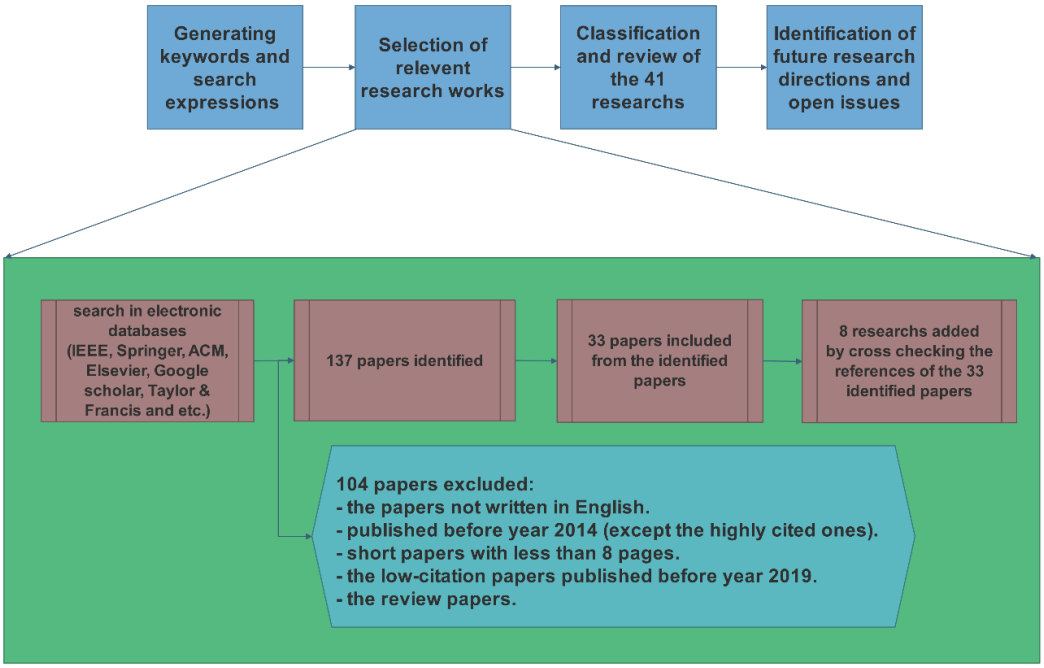

Fig. 4: Summary of research methodology.

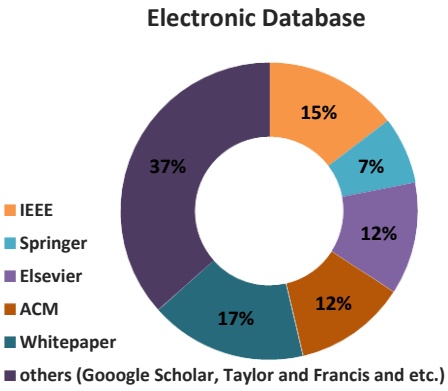

Fig. 5: Distribution of research works based on electronic database.

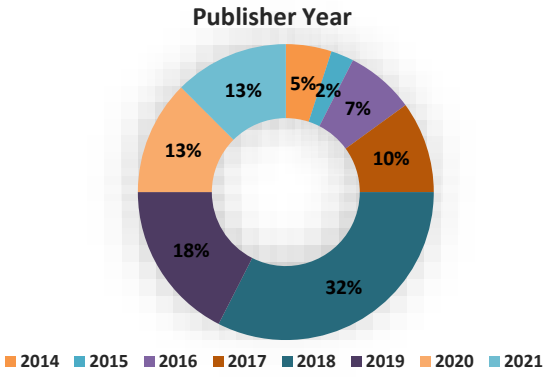

Fig. 6: Distribution of research works based on publish year.

Scalability Issue

With the continuous growth of blockchain systems, scalability is emerging as a challenging issue and the biggest barrier to the widespread adoption of the blockchain.

Despite the increasingly growing scale of the system, a scalable blockchain not only retains its functionality and performance but also takes advantage of the larger scale system to improve its performance. Ever-increasing transaction data and number of the participating users in blockchain systems lead to scalability issues such as low

throughput (in terms of transactions per second), latency and also the greater need for storage and bandwidth. Since blockchain is a consensus-based and distributed data ledger where all the full nodes keep a copy of the data ledger and validate all the transactions, the more users join the network, the more time is needed to reach a consensus on the transactions. Therefore, latency increases and the overall throughput of the system decreases. Moreover, with the growing size of the blockchain, the full nodes require more storage space to replicate the data ledger and more bandwidth to download the whole data ledger to bootstrap at initialization time. In this survey, scalability solutions are grouped into five categories as follows:

- On-chain scalability
- Off-chain scalability
- Scalable consensus mechanisms
- Directed Acrylic Graph (DAG)-based scalability
- Horizontal scalability through sharding

Fig. 7 demonstrates the taxonomic categories and the existing solutions in each category that have been discussed in this paper. Each category of scalability solutions can make changes on the different layers in the blockchain architecture. In the following sections, a detailed survey of the existing scalability solutions is presented. Meanwhile, these solutions are analyzed from the perspectives of scalability metrics including throughput, latency, storage, and bandwidth.

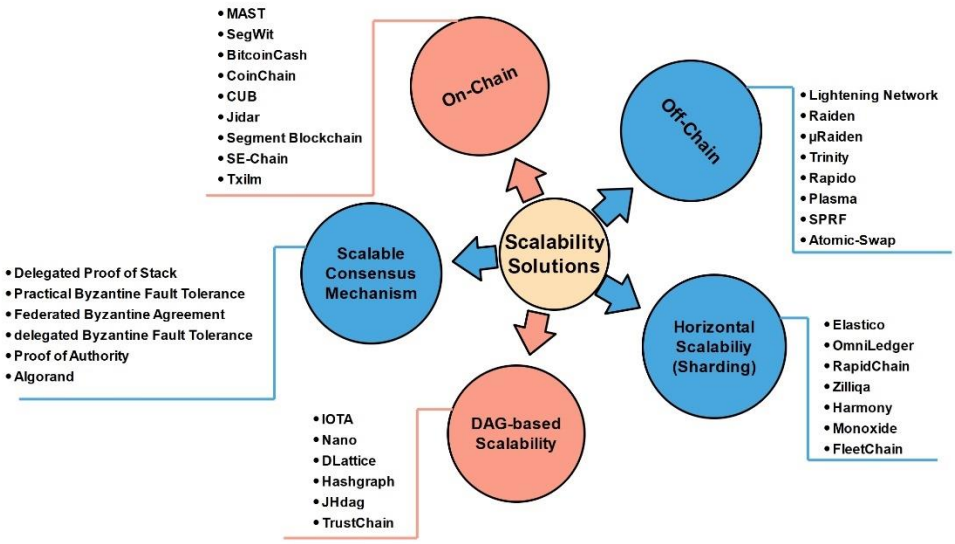

Fig. 7: Taxonomy of scalability solutions.

L. On-Chain Scalability Solutions

On-chain scalability solutions refer to the solutions that modify some key elements of the data layer in the blockchain architecture. According to the Data layer section of this paper, key elements of the data layer are block, transaction, Merkle tree, digital signature, and hash function.

For example, some works [31], [100] use a bigger block method in which the block's size is increased to a larger size. The bigger block method provides a higher throughput because bigger blocks can contain more transactions, thus whenever a block is added to the blockchain more transactions are confirmed, although the bigger blocks lead to a higher block propagation delay.

In addition, block compression is another method that is used by some works [101], [102] to save both the space of the blockchain and the bandwidth of the network. In this section, some of the important on-chain scalability solutions are introduced.

I) Merkelized Abstract Syntax Tree (MAST)

Merkelized Abstract Syntax Tree (MAST) [29], as an addition to Bitcoin, has been proposed to improve the scalability of the blockchain from the aspect of the capacity of the Bitcoin scripting. To do so, it combines the Merkle Tree and Abstract Syntax Tree (AST) concepts to represent the script parts of the Bitcoin transactions, both compactly and securely. Indeed, the transaction outputs in Bitcoin include a locking script also known as "encumbrance" that specifies the conditions under which the recipient can spend that output. Furthermore, the AST is a tree that represents the abstract syntax of the source code of a computer program as a hierarchical tree structure. Therefore, by employing AST, MAST is able to store the more complex locking scripts in Merkle tree format and remove unused parts of a script from the transaction. Thus, MAST combining the Merkle tree and AST, provides both data integrity and transaction compression. To sum up, MAST causes smaller

transactions and more privacy, and allows larger smart contracts, however, it increases the complexity of the contracts agreed to the Bitcoin.

II) Segregated Witness

Segregated Witness (Segwit) [30] is a modification to Bitcoin [74], whose solution for scalability is to free up space on the blocks in order for more transactions can be included in a block, whereby more transactions would be carried out and transaction confirmation speeds up.

Bitcoin transactions consist of inputs, outputs and also witness data including signature and script for transaction validation. To free up space on the block, Segwit removes witness data from the Bitcoin transactions and stores it onto a separate block instead of maintaining it on the blockchain. Since witness data consumes nearly 70% of the block size, by removing it, Segwit can process 1.7 to 4 times more transactions than Bitcoin, resulting in reduced transaction fees [103].

Segwit also aims to prevent transaction malleability in Bitcoin. Malleability refers to the problem that allows attackers to change the transaction ID of an existing transaction by modifying its digital signature, while its data is the same as the original one, and then rebroadcast it onto the network making other nodes think the original one has not been confirmed [104]. In addition to solving scalability and malleability issues, Segwit paves the way for developing off-chain solutions (e.g. Lightening Network [39] which will be discussed in the following sections) and now is utilized by Litecoin [83]. Having said all these advantages, the main disadvantage of Segwit is that, as a soft fork, it leads to a fungibility problem because there is no need for all the nodes in the network to upgrade the older version of Bitcoin. Additionally, Segwit extremely increases resource usage such as capacity and bandwidth because it needs to process more transactions at the same time. Furthermore, the implementation of Segwit is challenging since it increases code complexity.

III) BitcoinCash

Unlike Segwit soft work, BitcoinCash [31], [32] is a hard fork from Bitcoin that splits the Bitcoin network into two new blockchains. To improve scalability, Segwit tries to reduce transaction size while BitcoinCash tries to increase the block size. In fact, BitcoinCash changes the native Bitcoin codebase to increase the block size limit from 1MB to 8MB resulting in a higher throughput that is averagely 116 transactions per second. Consequently, BitcoinCash enables faster transaction processing than the Bitcoin network, while at the same time, it compromises decentralization because fewer nodes can process or propagate the larger blocks. In other words, it requires more processing capacity and bandwidth.

IV) CoinChain

CoinChain [33] is a scalable and prunable blockchain while keeping privacy and works as a sidechain for Bitcoin. Coinchain is scalable from the perspective of storage and blockchain size, whereas its transaction throughput is the same as Bitcoin. Indeed, privacy concerns to provide anonymous transactions and preserve confidentiality and anonymity of the sender, receiver and amount of the transactions, make the existing cryptocurrencies more complicated and restrain blockchain pruning. On the contrary, CoinChain is a straightforward and simple protocol that operates like physical cash transfer systems where banknotes with unique serial numbers, corresponding to distinct denominations, are transacted between users. Therefore, each coin is identified by a unique CoinID and coin ownership is transferred through transactions. Consequently, blockchain can be pruned by just keeping the last owner of the coins. Nevertheless, there are some shortcomings regarding CoinChain. First, fractional amount payment is not allowed. Second, the users are required to mix or spend out all the coins they pegged in the first place to ensure privacy. Third, in CoinChain, auditing for different purposes such as tracking money laundering or tax evasion, is feasible only via full disclosure of the transaction information.

V) Storage efficient solutions

High storage usage is a challenging issue that restricts many devices to participate in the blockchain because of storage space limitations. In this section, some works focusing on storage optimization in the blockchain will be discussed.

CUB [34], to reduce storage usage, splits the entire network into smaller units namely "consensus units" where the nodes cooperatively store one copy of the blockchain instead of each node keeping its own distinct copy. Therefore, it helps to save storage space for blockchain network peers. In addition, CUB provides solutions to optimize the block assignment and minimize the query cost. The main drawback of CUB is that it relies on a strong trust assumption which is hard to satisfy in practice.

Jidar [35] is a data reduction strategy without trust assumption for Bitcoin in which each node only has to store relevant transactions that it cares about, besides the branches of the Merkle tree from the whole block that is needed for the validation of new transactions. Jidar is able to reduce the storage cost of each node by about 1.03% compared with the native Bitcoin system. The bad thing about Jidar is that it does not support general-purpose smart contracts.

In addition, if some nodes require to have a whole block, they first need to query the pieces of the block data from different nodes and then cohere all pieces into a block, however, this functionality requires an incentive mechanism to be added.

Segment blockchain [36] is a data-reduced storage

approach for blockchain. The main idea of the Segment blockchain mechanism is that it partitions the blockchain into segments and then allows each node to store only one segment of the blockchain rather than the whole blockchain. It is proved that Segment blockchain reduces storage requirements significantly without compromising either the security or the decentralization of the blockchain. Furthermore, Segment blockchain facilitates blockchain sharding because it separates transaction verification from transaction storage. On the downside, it is only suitable for applications that do not need a large transaction output. SE-Chain [37] is also a scale-out blockchain framework that enhances storage scalability. In the data layer of the SE-Chain framework, each transaction is stored in the Adaptive Balanced Merkle tree (AB-M tree) and the full nodes store a part of the blockchain designated by the duplicate ratio regulation algorithm. In addition, to ensure the safety of the stored data on the full nodes, a node reliability verification method is presented. Another contribution of the SE-Chain is that it provides fast and efficient data retrieval using the AB-M tree.

VI) Block compression

Some works in the literature have used block compression to save the network bandwidth, an important factor that impacts blockchain scalability.

One such solution is Txilm [38] in which each block includes a list of compact presentation of the transactions rather than the original transactions. To produce a compact of a transaction, the transaction is hashed twice, first using SHA256 which generates a hash of 256-bits so-called TXID, then using a hash function (e.g. CRC32, CRC4p or CRC64) which generates a k-bit small-sized hash value so-called TXID-HASH. Therefore, the final output, i.e. TXID-HASH, is the compact presentation of the transaction that is included in a block along with the TXID-HASH of other candidate transactions and the block header which includes SHA256 Merkle root of all containing TXIDs. After that, the resulting compact block is propagated into the network by the user. Once receiving the compact block by full nodes, they should search into their memory pool to find a TXID matched with each TXID-HASH listed in the compact block. If one matched TXID is found, The TXID-HASH will be accepted. Otherwise, the full node requests the sender or other nodes for the missing TXID. Moreover, the hash collision that happens whenever multiple matches are found for a TXID-HASH, is resolved using Merkle root. As a final point, Txilm results in 80 times data reduction, thus saving the network bandwidth considerably and improving the blockchain throughput.

Comparison of On-Chain Scalability Solutions

To give a clear overview of the on-chain scalability solutions, a comparison of them is summarized in Table 3 where the mechanism used by each solution to improve scalability, and also the scalability metrics over which they achieve improvement are specified. Moreover, the advantages and disadvantages of each solution are neatly summarized.

Table 3: Comparison of on-chain scalability solutions

Solution	Mechanism	Scalability measurements				Advantages	Disadvantages
		Throughput	Latency	Storage	Bandwidth		
MAST [29]	Transaction compression	--	--	Low↓	--	- Smaller transactions - More privacy - Larger smart contracts	- Increases complexity of permitted contracts - Not complete privacy
SegWit [30]	Increased block capacity	High↑	--	--	High↑	- High transaction speed - Paves the way for developing off-chain solutions	- Results in fungibility problem - Increases processing capacity and bandwidth usage - Difficult to implement
BitcoinCash [31], [32]	Increased block size	High↑	--	--	High↑	- Increases throughput	- Compromises decentralization
CoinChain [33]	Blockchain Pruning	--	--	Low↓	--	- Simple and easily understandable - Secure with high privacy	- Fractional amount payment is not allowed - Users need to mix or spend out all the coins they initially pegged - Full transaction disclosure is needed for auditing
CUB [34]	Saving storage usage	--	--	Low↓	--	- Storage efficient	- Relies on a strong trust assumption
Jidar [35]	Data reduction	--	--	Low↓	--	- Storage efficient	- Does not support the general-purpose smart contracts
Segmant Blockchain [36]	Data-reduced storage	--	--	Low↓	--	- Storage efficient	- Suitable for the applications that do not need a large transaction output
SE-Chain [37]	Storage scalability	--	--	Low↓	--	- Efficient storage and data retrieval	--
Txilm [38]	Block compression	High↑	--	--	Low↓	- Saves bandwidth - Increases throughput	--

M. Off-Chain Scalability Solutions

The off-chain scalability refers to the solutions in which some portion of the transactions are offloaded from the blockchain to ease the burden of storing all the blockchain data in the main chain. Indeed, the off-chain transactions are executed outside the blockchain and only the final states are to be applied in the main chain. Consequently, they mitigate the issues arising from the ever-growing of the blockchain data thus improving the scalability and overall performance of the blockchain. Additionally, the off-chain transactions lead to lower fees and almost zero waiting time.

The off-chain mechanisms are usually in the form of payment channels [39]-[42] and sidechains [44]. A payment channel allows users to interact and transact with each other without using the expensive and slow blockchain and then broadcast the final closing transaction into the blockchain network to update their states. The payment channel is also called the state channel because it modifies and maintains the states of the main blockchain and then applies the last state to the main chain. On the other hand, a sidechain is an individual blockchain linked to its parent blockchain using a two-way peg [105] that allows users to interchange their assets between the sidechain and the parent chain at a prefixed rate.

In the following, some of the off-chain solutions that have attracted more attention are introduced.

I) Lightning network

The Lightening Network of Bitcoin [39] is one of the prominent examples of off-chain solution, which utilizes the payment channels to lighten the workloads of the main chain in Bitcoin. Therefore, every two users willing to transact with each other must first establish a channel between each other. For doing so, they first need to share a multi-signature address (wallet) and then they both deposit a certain amount of Bitcoin into that address. After that, they can do unlimited payment transactions between each other quickly and with minimal fees. After the transactions, the payment channel is closed and the final transaction is broadcasted to the Bitcoin blockchain to update the balance of the two users. The final transaction charges a fee from the payer user.

Despite all these advantages, Lightening Network has some drawbacks as follows: (1) it is less secure than the original Bitcoin, (2) it also supports only the micropayments for Bitcoin, and (3) it forces the users interacting with each other to be online at the same time and follow the same payment path.

II) Raiden Network and µRaiden

Raiden Network [40] is the Ethereum version of the Lightening Network, which allows Ethereum users to open a private channel namely "state channel" under which they can perform off-chain transactions and transfer tokens immediately and economically. The Raiden network is based on the same concepts as Lightening Network, however, opposite to the Lightening Network, it supports general-purpose transactions as the Ethereum supports general-purpose smart contracts.

µRaiden [41] is the first release of the Raiden network launched on the Ethereum mainnet. µRaiden is a micropayment solution for fast and free ERC20 token exchange. It is a many-to-one unidirectional payment channel framework that does not allow multi-hop transfers through payment channels, while Raiden is a many-to-many bidirectional solution that enables multi-hop transfers via bidirectional payment channels. Consequently, Raiden has a more complicated design than µRaiden.

III) Trinity

Trinity [42] is analogous to the Lightening Network and Raiden and provides an off-chain scaling solution using state channel technology. The difference is that it is built on the Neo blockchain [52] and aims to achieve real-time payment with low fees and provide protection for Neo assets. It increases the throughput of Neo blockchain considerably.

IV) Rapido

Rapido [43] is a scalable blockchain that provides multi-path payment channels, whereas before-mentioned off-chain solutions [39]-[42] offer a single-path payment. Single-path payments are vulnerable to leakage of sensitive information like payment value and also may result in overload issues since in the case that payment value goes beyond the deposit of every single path between two involved users, all the pre-established payment channels have to be closed and a new one needs to be established. Hence, Rapido has been presented to address these two issues by proposing Value Distributing Problem (VDP) program, whereby the payment value is divided into two or more sub-values and then sub-payments are settled using different payment paths. Furthermore, Rapido introduces Distributed Hashed Timelock Contracts (DHTLC) to ensure the security of these sub-payments. Rapido yields a success rate over 3 times higher than Lightening Network [39], although due to the existence of several intermediaries in multi-path payments, the willingness of individual donation is diminished [106].

V) Plasma

Plasma [44] provides an off-chain scaling solution for the Ethereum network through a sidechain mechanism. It employs a hierarchical tree-like structure of the chains called "child chains", that stem from the Ethereum main chain as its root (as shown in Fig. 8 [44]). Each child chain is a smaller version of the main chain and has its own subtrees of child chains. The child chains are smart contracts built on the main chain and have their own set

of rules and operate independently and in parallel to other chains. Therefore, child chains can be utilized for different purposes and interact with each other.

That is to say, the states maintained by the child chains are updated in the main blockchain periodically and verified by validating Merkle root. Plasma also allows the users to transfer their assets to the main blockchain by executing an "output transaction".

Taking advantage of these characteristics, Plasma can reduce the congestion of the Ethereum blockchain considerably and result in fast and low-cost transaction processing. Another advantage of Plasma is that it is consistent with other scalability solutions such as sharding and big blocks.

On the downside, to guarantee immutability in all child chains, many security considerations should be considered and addressed by the Plasma framework. Another challenge in Plasma is that in the case that, at the same time, all the users decide to leave the child chain and transfer their assets to the main chain, processing all the requests is impossible for the main chain.

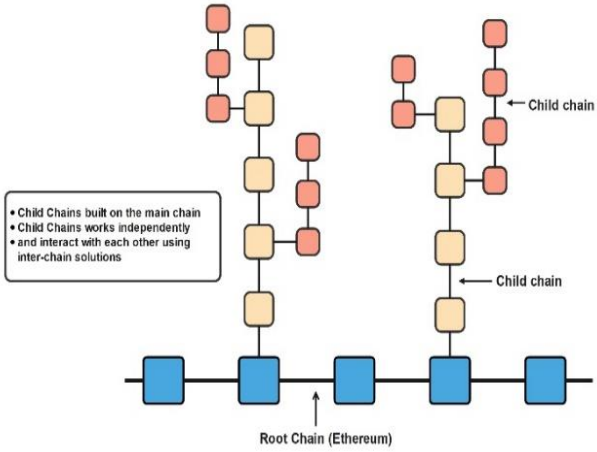

Fig. 8: Tree-like structure of Plasma blockchain [44].

VI) Smart Program Runner Framework (SPRF)

SPRF framework [45] is a sidechain implemented on the Stellar blockchain [50]. It is designed to move decentralized applications data off the blockchain and only stores the hash of the application state on the main chain. Therefore, it allows the state-tracking of smart programs on the blockchains. Indeed, it provides a platform including different applications, that enables decentralized compute-intensive software to be executed on the blockchain securely and efficiently. Another good thing about SPRF is that it can set up a sidechain for the existing blockchains without any need for revision.

It is important to highlight that there is another type of scalability solution named inter-chain which is similar to the sidechain. Inter-chain blockchains are used to connect different blockchains in a sidechain technology and solve interoperability problems between them. An inter-chain blockchain defines necessary protocols and standards enabling the blockchains to communicate among themselves. Atomic-swap [46] solutions are an example of inter-chain blockchain that provide an infrastructure for interacting between blockchains without the need for any centralized intermediaries, although they are applicable in bounded situations. For example, each blockchain must conform to extra programming features to be able to communicate with other blockchains.

Comparison of Off-Chain Scalability Solutions

In this section, a tabular comparison of the discussed off-chain scaling solutions is presented. As Table 4 demonstrates, these solutions have been investigated in terms of scalability improvements and grouped based on their off-chain mechanism. Moreover, their advantages and disadvantages have been outlined.

N. Scalable Consensus Mechanisms

There exist some scalable consensus mechanisms aiming to optimize the scalability and performance of blockchain systems. To achieve this goal, these mechanisms revolutionize the consensus layer to speed up the consensus-making process and subsequently increase the transaction throughput. In the Consensus layer section of this paper, a taxonomy of consensus algorithms and their underlying features has been presented. This taxonomy includes three types of consensus algorithms, namely Proof-based, Vote/ BFT-based and DAG-based.

DAG-based consensus algorithms used in the DAG-based data ledgers are potentially scalable, while two other types are not inherently scalable and scalability is a challenging issue in early consensus protocols developed based on them. Hence a variety of novel protocols have been proposed to renovate proof-based and vote/ BFT-based protocols.

For example, PoW is a primary proof-based protocol that has limitations regarding speed and scalability metrics such as throughput, latency, computational power and transaction capacity, although, it scales well in terms of network size and provides permission-less access to the network. Moreover, PoS was proposed as an alternative to PoW that mitigates some of its scalability issues such as high computational power, high latency and low throughput. On the other hand, BFT-based protocols have higher throughput than proof-based ones, but due to some internal drawbacks such as scalability and communication overhead, they are only suitable for a private network and need an identity management system.

Some proof-based consensus algorithms proposed in the literature are Proof of Stake (PoS), Proof of Authority (PoA) [53], [54], Proof of Elapsed Time (PoET) [107], Proof

of Capacity (PoC) [108], Proof of Importance (PoI) [109] and Proof of Burn (PoB) [110].

On the other side, Voting-based consensus algorithms are Delegated Proof of Stake (DPoS) [47], [48], Practical Byzantine Fault Tolerance (PBFT) [49], delegated Byzantine Fault Tolerance (dBFT) [51], [52], [111], Federated Byzantine Agreement (FBA) [50] and Algorand protocol [55]. In the following, some of the mentioned consensus algorithms are described briefly. Further details on consensus algorithms are available in [99].

I) Delegated Proof of Stack

Delegated Proof of Stake (DPoS) [47], [48] is an evolution of the PoS algorithm and allows the blockchain to reach consensus using a democratic manner. In DPoS, the stackers vote and elect delegates to validate the next block on their behalf. Delegates are also called validators or witnesses. During the voting process, stackers pool their stack into a staking pool and link them to a particular delegate. For each new block, between 20 and 100 delegates are chosen depending on the system. Chosen delegates add blocks to the chain in a Round-Robin manner. In fact, contrary to PoW and PoS which are competing systems, DPoS is a collaborative system where the delegates collaborate to make the blocks. The delegates that constantly miss their block or publish invalid blocks, will be voted out by stackers and replaced. The transaction fee for each validated block is shared between the stackers who elect the successful delegate. DPoS is partially centralized, however, it is more scalable than PoW and PoS. Furthermore, DPoS is more susceptible to 51% attack, because its consensus process depends on a small set of delegates.

EOS [85] is a blockchain technology, which utilizes DPoS to elect and schedule the block validators. Then, elected validators use an Asynchronous Byzantine Fault Tolerant (ABFT) consensus mechanism to validate and confirm the block proposed by the active validator and reach a consensus on it. EOS is aiming to enhance scalability and eliminate transaction fees. It also facilities the DApps development process. In addition to EOS, BitShares [48], Steemite [112], Ark [113], Cardano [114] and Lisk [115] are some of the well-known projects employing the DPoS consensus mechanism that is a suitable solution for the scalability problem.

Table 4: Comparison of off-chain scalability solutions

Solution	Mechanism	Scalability measurements				Advantages	Disadvantages
		Throughput	Latency	Storage	Bandwidth		
Lightening Network [39]	Payment channel	High↑	Low↓	Low↓	Low↓	- Instant transactions - Lower fees	- Less secure - Supports only bitcoin micropayments - Needs the users to be online at the same time and follow same payment route
Raiden [40]	Payment channel	High↑	Low↓	Low↓	Low↓	- Fast and free transactions - Enables multi-hop transfers - Supports general purpose transactions	- Implies more complexity because it allows multi-hop transfers
µRaiden [41]	Payment channel	High↑	Low↓	Low↓	Low↓	- Fast and free ERC20 token - Supports general purpose transactions	- Does not allow multi-hop transfers
Trinity [42]	Payment channel	High↑	Low↓	Low↓	Low↓	- Real-time payment - Low-cost transactions - Privacy protection	- Supports only payment transactions
Rapido [43]	Payment channel	High↑	Low↓	Low↓	Low↓	- Avoids overload issue - Prevents privacy leaking - Mitigates the skewness and congestion issue	- Discourages individual donation
Plasma [44]	Sidechain	High↑	Low↓	Low↓	--	- Hierarchical structure - Reduces the congestion of the main blockchain - Fast and low-cost transactions - No need to users be online at the same time	- Complicated to be implement - Long waiting time for transferring assets to the main chain
SPRF [45]	Sidechain	--	--	Low↓	--	- Secure and also computationally efficient - Applicable to the existing blockchain without any modifications	--
Atomic-swap solutions [46]	Inter-chain	--	--	--	--	- Solve interoperability between different blockchains	- Work under specific situations

II) Practical Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance (PBFT) [49] is a variation of the vote/BFT-based consensus algorithm, in which the nodes reach consensus using a collective decision-making strategy even if some nodes withhold responding or respond incorrectly. PBFT operates in successive rounds called views. Each view has a primary node called leader and other nodes are referred to as backup nodes. The leader is changed in every view. PBFT consensus rounds consist of three phases: pre-prepare, prepare and commit.

Fig. 9 [116] shows an illustration of the PBFT consensus algorithm. As is shown, in the pre-prepare phase, the leader multicasts the next record (block) to the backup nodes. Then, in the prepare phase, after receipt of the pre-prepare message, the backup nodes validate its veracity and multicast a prepare message to all the other nodes in the consensus group. After that, in the commit phase, upon receiving prepare messages from more than two-thirds of all the nodes, each backup node multicasts a commit message to the consensus group and then waits for more than two-thirds of commit messages, to ensure that the majority of nodes have come to the same decision. Consequently, all the honest nodes agree unanimously on the valid record. Although PBFT is energy efficient and increases the transaction rate, it suffers from high communication overhead. Therefore, it is not scalable enough to be used in public networks and thus is only applicable to private and permissioned networks. Moreover, PBFT mechanisms are vulnerable to Sybil attacks where an adversary takes over the network by creating multiple fake identities for malicious purposes. Hence, the PBFT mechanisms are usually used in combination with other mechanisms.

For example, Hyperledger Fabric [90], an open-source blockchain framework for developing blockchain-based applications, has utilized a permissioned version of PBFT. In addition, Zilliqa [66] is a high-throughput blockchain that uses PBFT for consensus-making together with PoW for establishing identities. Tendermint [117] is also a consensus protocol that merges PBFT with DPoS to bring PBFT to a public blockchain.

III) Federated Byzantine Agreement

Federated Byzantine Agreement (FBA) [50] is another BFT-based consensus mechanism operating based on "quorum" and "quorum slice" concepts. The quorum refers to the nodes that should reach a consensus on the information that is to be stored in the blockchain. The quorum consists of individual quorum slices that are subsets of the quorum nodes. The transactions are confirmed only if a required number of the quorum slices agree on it. Hence, the FBA data ledger can be updated without requiring all the nodes to agree, resulting in network scalability and fast transaction with low cost.

Stellar [50] and Ripple [84] are two main cryptocurrencies using the FBA consensus mechanism. Steller has implemented an enhancement of FBA. It provides an open membership so that anyone can join the network or even be a validator without the need to be verified ahead of time. In addition, in the Stellar network, users can determine which quorum slice they trust. On the other side, Ripple has a close membership and only pre-selected validators vote on the veracity of the transactions. Therefore, Steller is more decentralized than Ripple.

IV) delegated Byzantine Fault Tolerance

delegated Byzantine Fault Tolerance (dBFT) consensus algorithm [51] was first introduced by Neo blockchain [52], a smart contract platform that is often referred to as "Ethereum of China". Generally speaking, there are three types of nodes in dBFT, called speaker, delegate and common node. The common nodes are the ordinary token holders that vote to elect delegates. Delegates form a consensus group for BFT consensus. Then, a speaker is randomly chosen among the delegates. The transactions created by the common nodes are received by the speaker node. The speaker node validates the received transactions and then creates a block containing a number of valid transactions. After that, the speaker multicasts the new block to the delegates. Upon receiving the new block, the delegates validate it separately and respond to the speaker (same as the process used in the PBFT). If more than two-thirds of the delegates confirm the new block, it will be added to the chain. The Neo employing dBFT provides a high throughput, making it applicable to large-scale commercial applications. Neo was designed to digitize assets using smart contracts and enables users to trade their digitized assets by two types of tokens namely Neo and GAS. dBFT is also used by other blockchains such as ONT [118]. A major disadvantage of dBFT is that the delegates require to provide a real identity to be elected during the voting process.

V) Proof of Authority

Proof of Authority (POA) [53], [54] is an alternation of PoS that uses identity as a stack. In the PoA protocol, a number of trusted nodes called validators are responsible for validating the transactions. A leader is randomly selected from the set of all validators to add a new block to the chain. Any leader that does not perform appropriately will be voted out and replaced by other validators.

POA leads to a scalable and high-throughput blockchain but due to its identity-based and centralized nature, it is more applicable in the private blockchains than public ones. POA Network [119] is the first public Ethereum-based platform that has employed the PoA consensus mechanism. It provides an open-source framework for smart contracts. All validators within PoA Network are licensed by United States notaries, and their identities reference a public notary database.

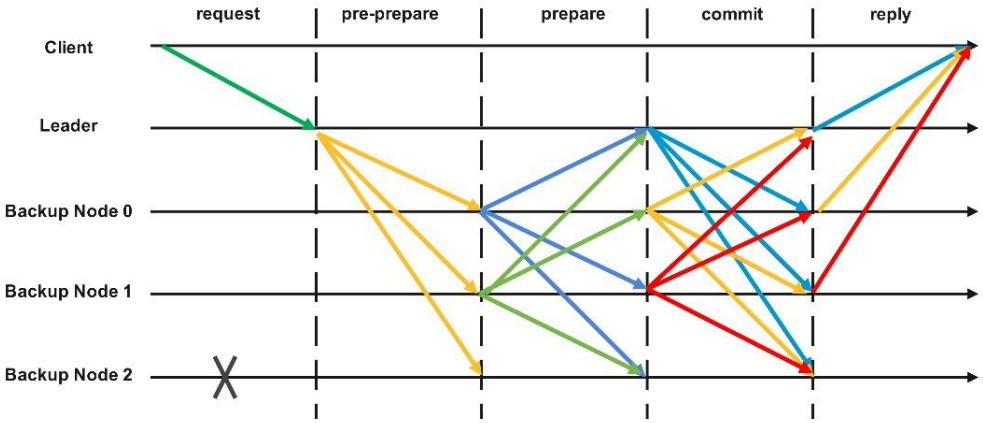

Fig. 9. Practical Byzantine Fault Tolerance (PBFT) protocol [116].

VeChain [120] is another blockchain project using PoA. In the VeChain network, the validators are called Authority MasterNodes (AMs) and to be an AM, users need to submit identifying information to the VeChain Foundation. VeChain is in an effort to enhance its PoA to provide a more randomized and distributed block-creating mechanism.

VI) Algorand protocol

Algorand [55] is a blockchain-based cryptocurrency that utilizes a new Byzantine Agreement (BA) called BA★ allowing users to achieve consensus on the next set of transactions with low latency. It is also able to scale consensus to millions of users. To achieve scalability, Algorand selects randomly a few representatives from the entire set of users. Representatives form a committee responsible for confirming transactions. to prevent Sybil attacks, Algorand assigns a weight to each one of the users based on the stack they own and as long as a weighted fraction of the users are honest, the consensus is guaranteed by BA★ protocol.

In addition, Algorand utilizes a novel mechanism based on Verifiable Random Functions (VRF) enabling it to choose committee members through a random and private way. Meaning that, by calculating a VRF of their private key and some public information in the blockchain, the participating users are able to individually specify whether they are selected to be on the committee or not. This prevents attackers to recognize the committee members ahead of time and plan a target on them. After computing VRF, only committee members have the right to propose a new block, thus they propagate the proposed block along with the VRF output which proves that the account is a committee member.

To achieve consensus using BA★ and ensure that all the nodes have the same view of the blockchain, the nodes will confirm the signature of the message containing proposed blocks and then, using the VRF proof, validate whether the proposer is a committee member or not. Next, through a cryptographic sortition, each node will compare the hash of the messages received from the committee members to identify the lowest one and then will only propagate the block proposal with the lowest VRF hash. Consensus Process consists of several interactive steps and continues until when a proposed block receives enough votes from weighted committee members. Therefore, it should be said that BA★ exploits a Pure PoS.

To sum up, Algorand is a fast, scalable and secure cryptocurrency. Moreover, it supports smart contracts and all kinds of financial transactions and programs, however, it still has not been adopted widely in the world of cryptocurrencies.

Comparison of Scalable Consensus Mechanisms

Table 5 provides a plain comparison between some common characteristics of the discussed scalable consensus mechanisms to help better understand their contributions.

O. DAG-based scalability solutions

Directed Acrylic Graph (DAG)-based data ledgers are an alternative to blockchain-based data ledgers, being a potential solution to address scalability issues. A DAG-based ledger is a network of individual transactions in which there are no blocks of the transactions and competition for appending new blocks, thus the confirmation time of the transactions is not bound to the interval of the blocks in the blockchain and transactions are processed independently. Consequently, the throughput of the data ledger is improved notably. Furthermore, DAGs alleviate transaction fees because the DAG-based consensus algorithms are simpler than the

Table 5: Comparison of scalable consensus mechanisms

Solution	Technique	Existing Projects	Scalability measurements				Advantages	Disadvantages
			Throughput	Latency	Storage	Bandwidth		
Delegated Proof of Stack (DPoS) [47], [48]	A variation of PoS combined with voting mechanism	EOS [85], BitShares [48], Steemite [112], Ark [113], Lisk [115], Cardano [114]	High↑	Low↓	--	--	- Scalable and fast - Better distribution of rewards - Energy efficient	- Partially centralized - Less secure against 51% attack
Practical Byzantine Fault Tolerance (PBFT) [49]	BFT- based	Hyperledger Fabric [90], Zilliqa [66], Tendermint [117]	High↑	Low↓	--	High↑	- High transaction rate - Energy efficient	- High communication overhead
Federated Byzantine Agreement (FBA) [50]	BFT- based and quorum-based	Stellar [50], Ripple [84]	High↑	Low↓	--	--	- Fast transaction - Low cost	--
delegated Byzantine Fault Tolerance (dBFT) [51], [52]	BFT-based	Neo [52], ONT [118]	High↑	Low↓	--	--	- High transaction throughput - Low latency - Energy efficient	- There is no anonymity and delegates need a real identity
Proof of Authority (PoA) [53], [54]	An alternation of PoS (identity as stack)	PoA Network [119], VeChain [120]	High↑	Low↓	--	--	- Scalable and high-throughput - Energy efficient	- Has an identity-base and centralized nature - More suitable for private blockchains
Algorand [55]	Pure PoS	--	High↑	Low↓	--	--	- High transaction throughput, par with large payment and financial networks - Scalable to many users - Low latency - Low transaction fee - Secure against DOS and Sybil attacks	- Not adopted widely

ones used by traditional blockchains. Generally, DAG-based solutions focus on both the data layer and consensus layer, to improve scalability. To do so, they adopt a novel data structure based on DAG and employ a consensus mechanism convenient to it.

In the following, some of the prominent DAG-based scalability solutions are discussed. In addition to the works that will be explained in this section, there are several other DAG-based data ledgers in the literature such as ByteBall [121], Spectre [122], GraphChain [123], Phantom [124], CDAG [125], Conflux [126], Dexon [127], Teegraph [128] and so on. More details about these and other DAG-based data ledgers are available in [98].

I) IOTA

IOTA [56] is the most popular DAG-based data ledger that has initially been designed for the Internet of Things (IoT) industry. IOTA is based on a data structure named Tangle. Tangle [76] is a particular type of DAG, made up of transactions that are connected by edges. Each edge from transaction A to transaction B indicates that transaction A validates and approves transaction B. An illustration of a Tangle had been shown before in Fig. 2.

In IOTA, before sending out a new transaction, users need to solve a simplified PoW problem and then validate two previous transactions simultaneously. Each user technically acts as a miner that mines the previous transactions to be able to send new transactions. Therefore, there is no transaction fee in IOTA. In addition, since the previously added transactions are validated by the new transactions, the more transactions are created by the users, the more transactions are confirmed per second. Subsequently, the throughput and confirmation time of the transactions are improved. Another advantage of the IOTA is that it provides security against quantum computers as it uses hash-based signatures rather than elliptic curve cryptography. On the other side, one of the disadvantages of IOTA is that it lacks smart

contracts, thus developing DApps on the IOTA is almost impossible.

II) Nano

Nano [57] uses a novel form of DAG namely Block-lattice. The Block-lattice architecture is a hybrid between blockchain and DAG and is made up of account-based blockchains. This means that each user holds a separate blockchain for each account, that represents the transaction history of that account. The participating users can only control and update their own individual blockchain with their private keys. Meanwhile, they update blockchains owned by other accounts asynchronously, rather than forming an agreement on a shared data ledger. Every transfer of nano coins requires two separate transactions/blocks, a send transaction deducting the amount from the sender's balance and a receives transaction adding the amount to the receiver's balance (see Fig. 10 [57]). The send transaction is signed by the sender account and is stored on the account chain of the sender while the receive transaction is signed by the receiver account and is stored on the account chain of the receiver. Furthermore, each transaction contains the current balance of its owner account.

Nano utilizes a variation of Delegated Proof-of-Stack (DPoS) consensus mechanism called Open Representative Voting (ORV) under which account holders can choose a representative to vote on their behalf regarding the validity of the transactions, even when the delegating account is itself offline. One drawback of Nano is that it is prone to spam attacks, meaning that it can be flooded with spam transactions, causing some valid transactions to be obstructed and network nodes to be out of sync.

III) DLattice

DLattice [58] is a permission-less blockchain with a double-DAG architecture under which Dlattice provides data protection and tokenization. DLattice has a double-DAG structure because each account has its own account-DAG and all account-DAGs form a greater DAG namely Node-DAG. Node-DAG organizes all the Account-DAG in the form of a Merkle Patricia Tree (MPT) using a Genesis header. Each Account-DAG structure consists of a token-chain and a data-tree. The token-chain is a unidirectional chain that records the income and expenditure history of the digital assets sent by the account, whereas, data-tree is a Red-Black Merkle Tree [129] combined with token-chain, that stores the digital fingerprint of the data asset and corresponding access control permissions. DLattice isolates transaction processing within each Account-DAG. Therefore, the transactions of the accounts can be processed in parallel resulting in fast transactions with minimal overhead. Instead of executing consensus at a fixed interval, DLattice uses a new DPoS-BA-DAG (PANDA) protocol to reach a low latency consensus among users only when the forks are observed. There are also some issues against DLattice. For example, it does not support smart contracts and also it is prone to DDoS attacks focusing on flooding false transactions and attacking smart contracts.

IV) Hashgraph

Hashgraph [59] is a DAG-based consensus algorithm based on a gossip protocol. In gossip protocol, each participating node randomly communicates with other nodes in the network to inform them about all the information it has, until the whole network is aware of all the transactions that have been processed so far.

In fact, nodes gossip about gossip. This means that they not only gossip about transactions but also gossip about the information that they have received from other nodes. Each member in the network maintains a separate chain to record the history of all the gossip events during which it receives some information. As it can be seen in Fig. 11 [59], the network members will eventually build a full history and create collaboratively a Hashgraph of all the gossip events. Then, each event in the Hashgraph is validated during a conventional Byzantine Fault Tolerance (BFT) consensus procedure. Hashgraph also enables visual voting, meaning that if two nodes have the same Hashgraph, rather than sending a vote message they can calculate each other's vote. Hence, the Hashgraph algorithm has very little communications overhead. Totally, Hashgraph-based communication patterns result in fast convergence of the information at all the nodes. All these features make Hashgraph a fair, fast and Byzantine Fault Tolerant solution, however, one drawback of the Hashgraph is that it is not secured against Sybil attacks thus it is more suitable for a permissioned network.

V) JHdag

JHdag [60], [61] is a PoW-based consensus mechanism that is designed on a novel DAG structure under which the network members can reach consensus at a large scale. In this structure, each block only contains one transaction in order to save network bandwidth when broadcasting blocks. Furthermore, the PoW puzzle is simplified to scale up processing capacity. Additionally, a mempool transaction assignment mechanism is designed based on the DAG structure to reduce the probability of processing a transaction by multiple miners, and hence reduce the waste of the capacity. To reach a consensus, a Nakamoto chain is embedded in a DAG structure that is strongly connected and incorporates miner information. There exist two types of blocks: regular blocks for carrying transactions and milestone blocks for making decisions. Blocks on the embedded Nakamoto chain are milestones

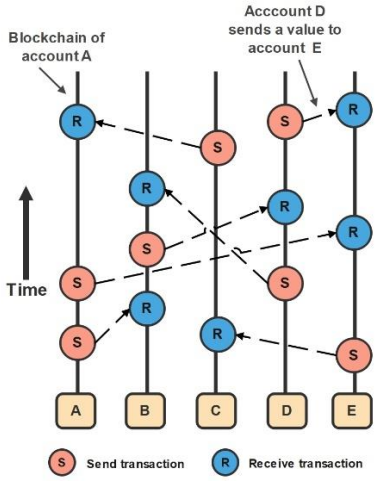

Fig. 10: Block-Lattice structure in Nano blockchain [57].

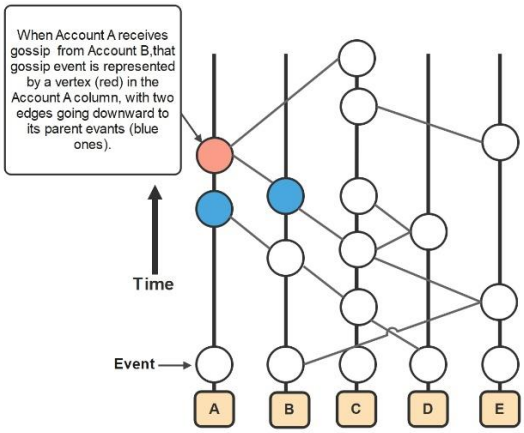

Fig. 11: Hashgraph, a full history of all gossip events [59].

and are harder to mine than regular blocks and each milestone block can verify multiple regular blocks.

VI) TrustChain

TrustChain [62] is a scalable, tamper-proof and Sybil-resistant blockchain, working based on the notion of trust between nodes. Each node has its own temporally ordered chain of blocks containing transactions that it participates in. Each block in TrustChain includes at most one transaction signed by both transacting parties and is linked back to the last block in the chain of both participating nodes using hash values. Accordingly, each block has two incoming and two outcoming pointers. As a result, transactions are arranged in a trusted and tamper-proof manner and form a DAG structure capable of creating trusted transactions without the need for any central authority or global consensus. In addition, to avoid Sybil attacks, TrustChain offers a new algorithm named NetFlow to calculate the trustworthiness of the network nodes according to the TrustChain graph. Indeed, the trustworthiness score which is assigned to each node in the graph determines if that node can contribute back the resources that it needs. Consequently, the network is affected only by the nodes with a positive score. Having said that, the transaction throughput of Trustchain is approximately 210 transactions per second which is not very high compared to centralized payment systems and some of the other scalable blockchains.

Comparison of DAG-based Scalability Solutions

In Table 6, a comparison of DAG-based solutions is presented. The promise of DAG-based solutions is to parallelize transaction processing whereby a high scalability is achieved in many aspects e.g. throughput and latency. For this purpose, they alter the data structure in the data layer of traditional blockchain and usually run a different algorithm in the consensus layer. To provide a clear vision, Table 6 summarizes some of the characteristics of the described DAG-based solutions, such as their data structure and consensus mechanism and also specifies their influence on the scalability measurements.

P. Horizontal Scalability Through Sharding

Sharding is a horizontal scaling solution in which adding more nodes to the network increases system performance. It refers to the techniques that partition blockchain nodes into subsets called shards. The workload of the blockchain is also distributed among the shards acting in parallel, leading to a high-performance and high-throughput blockchain. In addition to horizontal scaling, there are few works towards vertical scaling of the nodes, e.g. Ostraka [130], in which instead of partitioning nodes in a blockchain system into shards, each node itself is sharded into multiple Node-Shards.

Fig. 12 illustrates an example of sharding architecture, where the transactions are distributed among multiple shards and processed in parallel. In the following, some of the sharding-based blockchains and their properties are briefly explained.

I) Elastico

Elastico [63] is the first sharding-based protocol for permission-less blockchains in which nodes have no pre-published identities. Elastico breaks up the network into multiple committees each of which handles a disjoint set of transactions. A “consensus committee” is responsible for combining agreed transaction sets of other committees. Elastico uses a PoW mechanism to establish identities and map them randomly to the committees and also uses a PBFT mechanism to reach consensus within each committee. The protocol proceeds in epochs. At the beginning of each epoch, the identities are re-established and the committees are reconstructed.

Table 6: Comparison of DAG-based scalability solutions

Solution	Structure	Mechanism	Consensus	Scalability measurements				Advantages	Disadvantages
				Throughput	Latency	Storage	Bandwidth		
IOTA [56]	Tangle	--	Cumulative weights of all transactions that directly or indirectly approve the transaction	High↑	Low↓	--	--	- Fast confirmation - No transaction cost - High throughput and low latency - Quantum resistant	- Does not support smart contracts yet
Nano [57]	Block-Lattice	Independent nonshared blockchains	ORV	High↑	Low↓	--	--	- Fast confirmation - High throughput and low latency - Fee-less	- Prone to spam attack
DLattice [58]	Double-DAG	--	PANDA	High↑	Low↓	--	--	- Fast confirmation - No transaction cost - High throughput and low latency - Provides data tokenization	- Prone to DDoS attack - Does not support smart contracts
Hashgraph [59]	Hashgraph built of all the gossip events	Gossip about Gossip/ event-based	BFT	High↑	Low↓	--	Low↓	- Fair, fast and Byzantine Fault Tolerant - Visual voting	- Not secured against Sybil attacks
JHdag [60], [61]	Embedded Nakamoto chain inside DAG structure	Flexible-PoW	PoW	High↑	Low↓	--	Low↓	- Little or no transaction fee - Save bandwidth - Concentration of mining power within mining pool - Reduces waste of capacity	--
TrustChain [62]	A chain of trusted transactions	Building trust between individuals	consensus is reached among transacting users without needing any global consensus	High↑	Low↓	--	--	- Sybil-resistant - Removes the requirement for global consensus	- not very high transaction throughput compared to centralized payment systems

Elastico can scale linearly with the number of participating nodes and it is efficient in terms of communication overhead. In addition, the network topology between honest nodes is connected and the communication channel is partially synchronous. In terms of resiliency, Elastico can tolerate malicious users controlling up to one-fourth fraction of the total computational power in the network, while resiliency for each committee is one-third of malicious processors. That is not good enough.

Although Elastico can improve scalability measurements such as throughput and latency considerably, it has some shortcomings as follows. First, it is not storage efficient as it needs all users to store the entire data ledger. Second, in order to reduce the communication overhead of running PBFT consensus, it needs to choose a small committee size that leads to a high failure probability. Third, Elastico does not guarantee the atomicity of the cross-shard transactions. Finally, the epoch randomness used by Elastico for establishing identities and formation of the committee is not fully bias-resistant and might be biased by malicious users.

II) OmniLedger

OmniLedger [64] is a secure and permission-less blockchain that provides scalability via sharding. To securely assign nodes to the shards, OmniLedger implements sharding using the Randhound protocol that provides a bias-resistant decentralized randomness. In order to process inter-shard transactions, shards run ByzCoinX, an enhancement of PBFT-based consensus in ByzCoin [131], that improves performance and robustnes

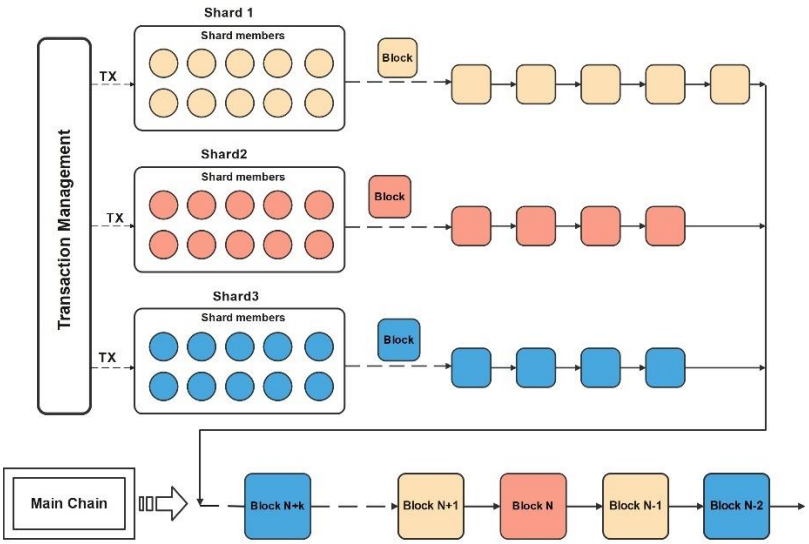

Fig. 12: Architecture of sharding-based blockchain.

against DoS attacks. Like Algorand [55], OmniLedger uses VRF and cryptographic sortition to pick a subset of the validators based on some per-validator weight functions. OmniLedger has also proposed the Atomix protocol that ensures atomically processing of cross-shard transactions. To optimize storage and reduce update overhead, OmniLedger uses the state blocks that provide checkpoints for the data ledger. In addition, OmniLedger provides low latency for low-value transactions using two-step trust-but-verify processing and also shows a latency of seconds for typical transactions.

Omniledger is a full-decentralization blockchain and has no single points of failure. In OmniLedger, the throughput increases almost linearly as the number of participating nodes increases. Furthermore, OmniLedger offers a throughput advantage par with centralized payment systems such as Visa, without compromising security or decentralization. It also is able to handle Visa-level workload.

On the other hand, similar to Elastico [63], OmniLedger can only tolerate up to one-fourth of malicious nodes in the entire network, and up to one-third of malicious nodes in each committee. In addition, per each confirmed block, the OmniLedger protocol needs to gossip multiple messages to all the nodes in the network. Another drawback of OmniLedger is that it needs the client to be an active participant in cross-shard transactions which is an inconvenient assumption for lightweight clients.

III) RapidChain

RapidChain [65] is the first one-third resilient sharding-based blockchain protocol scaling public blockchains via full sharding of computation, communication and storage. Meaning that, in addition to parallelizing transaction processing, the data ledger is also divided into partitions each of which is stored by one of the committees. Furthermore, to provide sharding of communication, RapidChain uses the Kademlia routing algorithm for committee-to-committee communication and cross-shard transaction processing. On the other hand, to process intra-committee transactions, each committee chooses a leader based on epoch randomness. The leader forms a new block and creates the block header and then propagates the block header using IDA-gossip protocol. Finally, the committee runs a synchronous BFT consensus protocol to make a consensus on the header of the block. Consequently, RapidChain needs only a sublinear number of bits to be exchanged per transaction.

To prevent Sybil attacks, RapidChain needs the nodes wanting to join the network to solve a PoW puzzle. All nodes solve PoW offline to avoid any interruptions in the protocol execution. In addition, to prevent a slowly-adaptive adversary from compromising one or more committees, RapidChain runs a reconfiguration protocol built on the Cuckoo rule [132] between epochs, without regenerating all the committees. In Rapidchain, total resiliency and committee resiliency are improved to one-third and one-two respectively. In addition, Rapidchain shows a much higher throughput and better latency than Elastico [63] and Omniledger [64].

IV) Zilliqa

Zilliqa [66] is a public blockchain platform designed to increase the transaction rate using sharding that enables parallel processing of transactions on multiple shards. Zilliqa also provides a smart contract platform and innovates a special-purpose smart contract language that follows a dataflow programming style that facilitates parallelizing of large-scale computation. Zilliqa uses PoW to establish node identities and prevent Sybil attacks. To reach consensus, Zilliqa uses an evolution of the PBFT

algorithm that is inspired from ByzCoin [131] and replaces the Message Authentication Code (MAC) used in the classical PBFT with a digital signature to lessen communication overhead to O(n) and also employs EC-Schnorr multisigniture to aggregate several signatures into an O(1)-size multisignature.

In addition, inspired from Bitcoin-NG [133], Zilliqa adopts two types of blocks: Transaction Blocks (TX-Block) and Directory Service Blocks (DS-Block). TX-Block includes the transactions sent by the users, whereas DS-Block includes metadata about the miners who participate in the consensus protocol. A TX-block will be finalized if it contains an EC-Schnorr multisignature by more than two-thirds of the miners.

Zilliqa reaches a great transaction rate, about a thousand times of Ethereum, although it shows the same local and global resiliency as Elastico [63] and Omniledger [64]. Zilliqa also suffers from some shortcomings. First, it does not provide storage sharding (state sharding). Meaning that all full nodes need to store and receive all the blocks and transactions, resulting in a high storage requirement. Second, Zilliqa is vulnerable to single-shard takeover attacks since it relies on PoW as a randomness generation mechanism.

V) Harmony

Harmony [67] is a fully scalable blockchain that similar to Rapidchain provides full sharding for transactions, communication and storage. To prevent Sybil attack and select validators, Harmony uses PoS rather than PoW, making it energy efficient. Consensus is reached using a new algorithm called Fast Byzantine Fault Tolerance (FBFT) which is linearly scalable in terms of communication complexity and is at least 50% faster than PBFT.

In FBFT, the leader executes a multi-signature signing process to collect the votes of the validators. Indeed, the sharding process is based on PoS and voting shares. Stackers gain voting shares proportional to their stack amount and then are randomly assigned to the shards. Like RapidChain [65], Harmony uses a cuckoo-based mechanism for resharding.

Harmony has proposed a unique algorithm for randomness generation by a combination of VRF and Verifiable Delay Function (VDF) that is unpredictable, unbiaseable, verifiable and scalable. In addition, Harmony uses RaptorQ function code to speed up the block propagation process within shards. It also uses an atomic locking mechanism to guarantee the consistency of the cross-shard transactions and adopts Kademlia as a routing mechanism for cross-shard communication and reducing communication overhead. The local (committee) and global resiliency of Harmony is the same as Elastico [63], omniledger [64] and Zilliqa [66].

VI) Monoxide

Monoxide [68] has been designed to scale out blockchain systems linearly without sacrificing security and decentralization. Towards this purpose, it divides the blockchain network into multiple independent and parallel instances called “consensus zones” and partitions the workload of computation, communication, storage and memory (for state representation) in the consensus zones. Each consensus zone has its own chain of blocks and runs the consensus process independently with minimized communication. In Monoxide, the blocks are created by miners. Although Monoxide uses PoW for mining blocks, its technique is impertinent to the consensus mechanism used per zone. To prevent attackers from controlling more than 50% of mining power in a single zone, Monoxide proposes a Chu-ko-nu mining mechanism that allows a miner to create multiple blocks in the different zones with one PoW solution. Consequently, the mining power is dispersed into multiple zones. since a PoW solution in Chu-ko-nu mining is more productive of blocks, it is more energy-efficient than traditional PoW. In addition, Monoxide uses eventual atomicity to ensure the atomicity of cross-zone transactions. In contrast to two-phase commit protocols that serialize the transactions, eventual atomicity has no additional delay and overhead.

VII) FleetChain

FleetChain [69] is a scalable and responsive blockchain with optimal sharding focusing on the intra-shard consensus and cross-shard transactions. To achieve an efficient intra-shard consensus, FleatChain proposes a Leader-Stable Fast Byzantine Fault Tolerance (FBFT) protocol that adopts a multi-signature schema to reduce message size during voting, combined with pipeline technology to enhance processing efficiency. Furthermore, for cross-shard transactions, a Responsive Sharding Transaction Processing (RSTP) protocol has been introduced that depends on the classical two-phase commit (2PC) protocol in which transaction inputs are locked/unlocked. Albeit unlike Omniledger protocol [64] where a client is considered as coordinator of the cross-shard transactions, in Fleetchain output shard leader operates as coordinator.

For intra-shard consensus, FleatChain utilizes a robust t out of u Multi-Signature Protocol with public key aggregation using Proof-of-Possession (PoP), shortly referred to as (t, u)-MSP-PoP, while uses (t, u)-AMSP-PoP (Aggregated Multi-Signature) for cross-shard transactions. To sum up, Fleetchain is scalable from the perspectives of computation (i.e., transaction throughput and latency), communication and storage, and its

scalability factor is $O(n/\log n)$ where n represents the network size.

Comparison of Horizontal Scalability/ Sharding Solutions

Table 7 and Table 8 provide a summarized comparison of the described sharding-based blockchains for a better understanding of their techniques and key features.

Future Directions and Open Issues

Although numerous researches addressing scalability challenges have been proposed in recent years, there are still some issues that were not resolved in the best way possible and were left open for future works. In this section, the open issues and future research directions of each category of scalability solutions are discussed separately.

Q. On-Chain Scalability

Secure blockchain pruning: large blockchain size leads to centralization problems due to limited storage capacity. An approach for reducing blockchain size is to remove non-critical and stale blockchain information to free up storage space on the nodes. Although a number of works [134] focusing on blockchain pruning have recently been proposed, there is still an outstanding question that needs to be answered: what data at what time must be removed so that security would not be compromised.

Blockchain data query: decentralization and data distribution in the blockchain lead to inconvenience for querying required data. As the blockchain grows, processing the various queries such as single, range and condition ones among a large amount of data, goes through performance and bandwidth issues. Hence, providing an efficient solution for querying blockchain data is an open issue that has not received enough attention in the literature.

R. Off-Chain Scalability

Future work can direct off-chain blockchains towards further off-chain computation techniques and conduct hybrid off-chaining mechanisms.

S. Scalable Consensus Mechanisms

Novel proof-based consensus mechanisms: Most scalable consensus mechanisms in the literature are based on voting, while there exist not many proof-based scalable ones. Therefore, designing secure and low latency proof-based algorithms is a topic that needs to be studied more. For example, an idea is to develop protocols that adopt non-transferable incentives such as reputation or familiarity, in which mining difficulty can be dynamically controlled.

Multi-block consensus mechanisms: redesigning consensus protocols so that they will be able to reach consensus on multiple blocks can improve throughput considerably.

T. DAG-Based Scalability

Trade-off: existing DAG-based systems failed to make a trade-off between multiple factors. For example, IOTA [56] and GraphChain [123] enhance performance and scalability while compromising security and consistency. On the other side, some DAG-based data ledgers e.g. Prism [135] and OHIE [136] provide strict consistency at the cost of scalability and performance. Hence, designing a DAG-based solution that can reach a balance between various metrics is still a challenging issue.

Supporting off-chain transactions: redesigning DAG-based systems for supporting off-chain transactions is an interesting direction for future work, which take advantage of both off-chain and DAG-based solutions.

System setup: setup configuration defines all the specific needs that must be available at the onset of the protocol to each participating node. Some existing DAG-based systems [56], [123], [137], [138] rely on the genesis block, whereas some others [57]-[59], [139], [140] initialize multiple parallel chains simultaneously, however, system setup using these parallel chains is unclear. Therefore, adopting a novel and transparent system setup can be considered as a future work.

U. Horizontal Scalability Through Sharding

Cross-shard transaction: cross-shard transactions lead to a lot of communication overhead and also reduce system performance and increase transaction confirmation time. Therefore, assigning transactions to different shards in a way that the cross-shard transactions be minimized is still an open issue. In this regard, the authors in [141] proposed a new sharding paradigm with optimal transaction placement, called OptChain. Furthermore, in [142] a new scalable permissioned blockchain named “Sharper” has been introduced that shards the transaction processing through clustering network nodes. Using two decentralized flattened consensus protocols, Sharper handles cross-shards transactions more efficiently. However, there is still a need for more efficient protocols for processing cross-shard transactions to reduce confirmation latency.

Resharding: resharding process is a challenging issue in sharding-based blockchains since it needs reshuffling the network which leads to huge data migration. SSChain [143] is the first public blockchain that provides full sharding with no reshuffling process and data migration.

Adaptive malicious attackers: resharding process is performed to prevent malicious users from overtaking a shard by corrupting the members of that shard during protocol epochs. Securing committee members against both slowly adaptive and fully adaptive attackers is a crucial problem that must be taken into consideration.

Table 7: Comparison of horizontal scalability solutions through sharding

Solution	Transaction Model	Identity Setup/ Committee Formation Mechanism	Intra-Consensus Mechanism	Cross-Shard Transactions Mechanism	Smart Contract	Total Resiliency	Committee Resiliency
Elastico [63]	UTXO	PoW	PBFT	Not supported	×	$1/4$	$1/3$
OmniLedger [64]	UTXO	RandHound	ByzCoinX	Sync, Lock/Unlock (AtomiX)	×	$1/4$	$1/3$
RapidChain [65]	UTXO	Offline PoW	Synchronous BFT	Sync, lock/Unlock	×	$1/3$	$1/2$
Zilliqa [66]	Account	PoW	An evolution of PBFT	not Supported	✓	$1/4$	$1/3$
Harmony [67]	Account	PoS	FBFT	Sync, Lock/Unlock	✓	$1/4$	$1/3$
Monoxide [68]	Account	Consensus zones (partitioning based on users address)	PoW (Chu-ko-nu mining)	Async, Lock-free (Eventual atomicity)	×	$1/2$	$1/2$
FleetChain [69]	UTXO	Proof of Possession (PoP) combined with PoW	Leader-Stable FBFT	Sync, lock/Unlock	✓	N/A	1/3

Table 8: Comparison of horizontal scalability solutions through sharding (tps: transactions per second, s: second, n: network size, m: committee size)

Solution	Sharding Components			Throughput *	Latency *	Communication Complexity
	Computation	Communication	Storage			
Elastico [63]	✓	×	×	48ktps	<900s	$O(m^2 + n)$
OmniLedger [64]	✓	×	×	28.8ktps	~100s	$O(log_2^m + n)$
RapidChain [65]	✓	✓	✓	128ktps	70s	$O(m^2 + mlog_2^n)$
Zilliqa [66]	✓	×	×	N/A	N/A	$O(n)$
Harmony [67]	✓	✓	✓	N/A	N/A	$O(log_2^n)$
Monoxide [68]	✓	✓	✓	1.23~2.56Mtps	23s	$O(m + n)$
FleetChain [69]	✓	✓	✓	N/A	N/A	$O(n/m)$

* The indicated throughputs and latencies are according to the evaluation of some sharding-based mechanisms conducted by Yu et al. in [97]. For more information about evaluation conditions, refer to Table 3 in [97].

An interesting idea is to incorporate the sharding process with machine learning algorithms to analyze the behavior patterns of users on the network and detect malicious users.

Conclusions

Scalability is the most important challenge to blockchain mass adoption.

This paper focuses on the blockchain scalability issue and reviews some related works in the literature dealing with it.

To do so, the scalability solutions are firstly classified into five categories including on-chain, off-chain, scalable consensus mechanism, DAG-based scalability, and sharding solutions.

Then, the key properties of these solutions along with their advantages and disadvantages are discussed to reveal their main contributions.

In addition, the discussed works are compared in terms of scalability improvements such as throughput, latency, storage and bandwidth. Finally, the future trends and open issues expected to be investigated through future works are discussed.

This paper provides a deep understanding of existing scalability solutions as well as the issues and challenges they deal with. Hence, it inspires novel ideas for more scalable and efficient blockchains in the future.

Credit Authorship Contribution Statement

Alemeh Matani: Writing- original draft, Investigation, Conceptualization. Amir Sahafi: Writing- review & editing. Ali Broumandnia: Writing- review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is affiliated with Islamic Azad University, South Tehran Branch, Tehran, Iran. Meanwhile, it did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Abbreviations

TPS	Transactions Per Second
DAG	Directed Acrylic Graph
UTXO	Unspent Transactions Output
PKI	Public Key Infrastructure
MPT	Merkle Patricia Trie
SPV	Simplified Payment Verification
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-curve Digital Signature Algorithm
BRS	Borromean Ring Signature
OTS	One-Time ring Signature
PoW	Proof of Work
PoS	Proof of Stack
P2P	Peer-to-Peer
BFT	Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance
VM	Virtual Machine
EVM	Ethereum Virtual Machine
DApps	Decentralized Applications
RQ	Research Question
MAST	Merkelized Abstract Syntax Tree
AST	Abstract Syntax Tree
AB-M	Adaptive Balanced Merkle
VDP	Value Distributing Problem
DHTC	Distributed Hashed Timelock Contracts
SPRF	Smart Program Runner Framework
PoA	Proof of Authority
PoET	Proof of Elapsed Time
PoC	Proof of Capacity
PoI	Proof of Importance
PoB	Proof of Burn
DPoS	Delegated Proof of Stake
dBFT	delegated Byzantine Fault Tolerance
FBA	Federated Byzantine Agreement
BA	Byzantine Agreement
VRF	Verifiable Random Functions
FBFT	Fast Byzantine Fault Tolerance
VDF	Verifiable Delay Function
RSTP	Responsive Sharding Transaction Processing
PoP	Proof-of-Possession

References

[1] Y. Chen, C. Bellavitis, "Blockchain disruption and decentralized finance: The rise of decentralized business models," J. Bus. Ventur. Insights, 13: e00151, 2020.

[2] L. Zhang, Y. Xie, Y. Zheng, W. Xue, X. Zheng, X. Xu, "The challenges and countermeasures of blockchain in finance and economics," Syst. Res. Behav. Sci., 37(4): 691-698, 2020.

[3] M. U. CHELLADURAI, S. Pandian, K. Ramasamy, "A blockchain based patient centric EHR storage and integrity management for e-Health systems," Heal. Policy Technol., 10(4): 100513, 2021.

[4] S. Shamshad, K. Mahmood, S. Kumari, C. M. Chen, "A secure blockchain-based e-health records storage and sharing scheme," J. Inf. Secur. Appl., 55: 102590, 2020.

[5] E. Bandara, D. Tosh, P. Foytik, S. Shetty, N. Ranasinghe, K. De Zoysa, "Tikiri—Towards a lightweight blockchain for IoT," Futur. Gener. Comput. Syst., 119: 154-165, 2021.

[6] U. Majeed, L. U. Khan, I. Yaqoob, S. M. A. Kazmi, K. Salah, C. S. Hong, "Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges," J. Netw. Comput. Appl., 181: 103007, 2021.

[7] P. Asghari, A. M. Rahmani, H. H. S. Javadi, "Internet of Things applications: A systematic review," Comput. Networks, 148: 241-261, 2019.

[8] P. Centobelli, R. Cerchione, P. Del Vecchio, E. Oropallo, G. Secundo, "Blockchain technology for bridging trust, traceability and transparency in circular supply chain," Inf. Manag., 59(7): 103508, 2021.

[9] B. Wang, W. Luo, A. Zhang, Z. Tian, Z. Li, "Blockchain-enabled circular supply chain management: A system architecture for fast fashion," Comput. Ind., 123: 103324, 2020.

[10] F. Luo, Z. Y. Dong, G. Liang, J. Murata, Z. Xu, "A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain," IEEE Trans. Power Syst., 34(5): 4097-4108, 2018.

[11] J. Wang, Q. Wang, N. Zhou, Y. Chi, "A novel electricity transaction mode of microgrids based on blockchain and continuous double auction," Energies, 10(12): 1971, 2017.

[12] M. Raikwar, S. Mazumdar, S. Ruj, S. Sen Gupta, A. Chattopadhyay, and K. Y. Lam, "A blockchain framework for insurance processes,"

in Proc. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS): 1-4, 2018.

[13] S. Gupta, A. Gupta, I. Y. Pandya, A. Bhatt, K. Mehta, "End to end secure e-voting using blockchain & quantum key distribution," Mater. Today Proc., 80: 3363-3370, 2023.

[14] X. Yang, X. Yi, S. Nepal, A. Kelarev, F. Han, "Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities," Futur. Gener. Comput. Syst., 112: 859-874, 2020.

[15] S. N. Mohanty et al., "An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy," Futur. Gener. Comput. Syst., 102: 1027-1037, 2020.

[16] Y. L. Gao, X. B. Chen, Y. L. Chen, Y. Sun, X. X. Niu, Y. X. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," IEEE Access, 6: 27205–27213, 2018.

[17] Y. Liu, G. Xu, "Fixed degree of decentralization DPoS consensus mechanism in blockchain based on adjacency vote and the average fuzziness of vague value," Comput. Networks, 199: 108432, 2021.

[18] N. Alzahrani, N. Bulusu, "Towards true decentralization: A blockchain consensus protocol based on game theory and randomness," in Proc. International Conference on Decision and Game Theory for Security: 465–485, 2018.

[19] E. Bandara, X. Liang, P. Foytik, S. Shetty, N. Ranasinghe, K. De Zoysa, "Rahasak—Scalable blockchain architecture for enterprise applications," J. Syst. Archit., 116: 102061, 2021.

[20] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, "LSB: A Lightweight Scalable Blockchain for IoT security and anonymity," J. Parallel Distrib. Comput., 134: 180-197, 2019.

[21] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, G. Danezis, "Chainspace: A sharded smart contracts platform," arXiv Prepr. arXiv1708.03778, 2017.

[22] A. I. Sanka, M. H. Chowdhury, R. C. C. Cheung, "Efficient high-performance FPGA-Redis hybrid NoSQL caching system for blockchain scalability," Comput. Commun., 169: 81–91, 2021.

[23] M. Muzammal, Q. Qu, B. Nasrulin, "Renovating blockchain with distributed databases: An open source system," Futur. Gener. Comput. Syst., 90: 105-117, 2019.

[24] Q. Qu, I. Nurgaliev, M. Muzammal, C. S. Jensen, J. Fan, "On spatio-temporal blockchain query processing," Futur. Gener. Comput. Syst., 98: 208-218, 2019.

[25] S. Linoy, H. Mahdikhani, S. Ray, R. Lu, N. Stakhanova, A. Ghorbani, "Scalable privacy-preserving query processing over ethereum blockchain," in Proc. 2nd IEEE Int. Conf. Blockchain, Blockchain 2019: 398–404, 2019.

[26] L. Zeng, W. Qiu, X. Wang, H. Wang, Y. Yao, Z. Yu, "Transaction-based Static Indexing Method to Improve the Efficiency of Query on the Blockchain," in Proc. 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA): 780–784, 2021.

[27] C. Riegger, T. Vinçon, I. Petrov, "Efficient data and indexing structure for blockchains in enterprise systems," in Proc. the 20th International Conference on Information Integration and Web-based Applications & Services: 173-182, 2018.

[28] V. Buterin, "Ethereum Whitepaper," 2013.

[29] J. Rubin, M. Naik, N. Subramanian, "Merkelized abstract syntax trees," XP055624837, Dec, 16(3), 2014.

[30] E. Lombrozo, J. Lau, P. Wuille, "Segregated witness (consensus layer)," Bitcoin Core Dev. Team, Tech. Rep. BIP, 141, 2015.

[31] Cash B. "Bitcoin Cash," 2017.

[32] M. A. Javarone, C. S. Wright, "From Bitcoin to Bitcoin Cash: a network analysis," in Proc. the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems: 77-81, 2018.

[33] W. K. Chan, J. J. Chin, V. T. Goh, "Simple and scalable blockchain with privacy," J. Inf. Secur. Appl., 58: 102700, 2021.

[34] Z. Xu, S. Han, L. Chen, "CUB, a consensus unit-based storage scheme for blockchain system," in Proc. IEEE 34th International Conference on Data Engineering (ICDE): 173-184, 2018.

[35] X. Dai, J. Xiao, W. Yang, C. Wang, H. Jin, "Jidar: A jigsaw-like data reduction approach without trust assumptions for bitcoin system," in Proc. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS): 1317-1326, 2019.

[36] Y. Xu, Y. Huang, "Segment blockchain: A size reduced storage mechanism for blockchain," IEEE Access, 8: 17434–17441, 2020.

[37] D. Y. Jia, J. C. Xin, Z. Q. Wang, H. Lei, G. R. Wang, "SE-Chain: A scalable storage and efficient retrieval model for blockchain," J. Comput. Sci. Technol., 36(3): 693-706, 2021.

[38] D. Ding, X. Jiang, J. Wang, H. Wang, X. Zhang, Y. Sun, "Txilm: Lossy block compression with salted short hashing," arXiv Prepr. arXiv1906.06500, 2019.

[39] J. Poon, T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[40] Network-Fast R. "Cheap, Scalable Token Transfers for Ethereum," 2018.

[41] "µRaiden. A Payment Channel Framework for Fast and Free Off-Chain ERC20 Token Transfers," 2018.

[42] "Trinity. Universal Off-Chain Scaling Solution for Neo," 2018.

[43] C. Lin, N. Ma, X. Wang, J. Chen, "Rapido: Scaling blockchain with multi-path payment channels," Neurocomputing, 406: 322-32, 2020.

[44] J. Poon, V. Buterin, "Plasma: Scalable autonomous smart contracts," White Pap., 1-47, 2017.

[45] M. Mallaki, B. Majidi, A. Peyvandi, A. Movaghar, "Off-chain management and state-tracking of smart programs on blockchain for secure and efficient decentralized computation," Int. J. Comput. Appl., 44(9): 822–829, 2022.

[46] R. van der Meyden, "On the specification and verification of atomic swap smart contracts," in Proc. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC): 176-179, 2019.

[47] Q. Hu, B. Yan, Y. Han, J. Yu, "An improved delegated proof of stake consensus algorithm," Procedia Comput. Sci., 187: 341-346, 2021.

[48] D. Larimer, "Delegated Proof of Stake (DPoS), Bitshare Whitepaper," 2014.

[49] M. Castro, B. Liskov, "Practical byzantine fault tolerance," in OSDI, 99(1999): 173-186,1999.

[50] M. Lokhava et al., "Fast and secure global payments with Stellar," in Proc. the 27th ACM Symposium on Operating Systems Principles: 80-96, 2019.

[51] M. Bareis, M. Di Angelo, G. Salzer, "Functional differences of neo and ethereum as smart contract platforms," in Proc. International Congress on Blockchain and Applications: 13-23, 2020.

[52] H. Da and E. Zhang, "Neo cryptocurrency," 2018.

[53] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," 2018.

[54] "Proof of Authority Whitepaper," 2018.

[55] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in Proc. the 26th symposium on operating systems principles: 51-68, 2017.

[56] M. Divya, N. B. Biradar, "IOTA-next generation block chain," Int. J. Eng. Comput. Sci, 7(04): 23823-23826, 2018.

[57] C. LeMahieu, "Nano: A feeless distributed cryptocurrency network," 16: 17, 2018.

[58] T. Zhou, X. Li, H. Zhao, "DLattice: A permission-less blockchain based on DPoS-BA-DAG consensus for data tokenization," IEEE Access, 7: 39273–39287, 2019.

[59] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," Swirlds Tech. Reports SWIRLDS-TR-2016-01, Tech. Rep, 34: 9-11, 2016.

[60] J. He, G. Wang, G. Zhang, J. Zhang, "Consensus mechanism design based on structured directed acyclic graphs," Blockchain Res. Appl., 2(1): 100011, 2021.

[61] G. Wang, J. Zhang, G. Zhang, J. He, "Consensus mechanism design based on structured directed acyclic graphs," arXiv, 2019.

[62] P. Otte, M. de Vos, J. Pouwelse, "TrustChain: A Sybil-resistant scalable blockchain," Futur. Gener. Comput. Syst., 107: 770-780, 2020.

[63] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, "A secure sharding protocol for open blockchains," in Proc. the 2016 ACM SIGSAC Conference on Computer and Communications Security: 17–30, 2016.

[64] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in Proc. 2018 IEEE Symposium on Security and Privacy (SP): 583–598, 2018.

[65] M. Zamani, M. Movahedi, M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in Proc. 2018 ACM SIGSAC Conference on Computer and Communications Security: 931–948, 2018.

[66] P. Barrett, "Technical Whitepaper," Zilliqa, 1-8, 2017.

[67] H. Team "Harmony, Technical Whitepaper," 2018.

[68] J. Wang, H. Wang, "Monoxide: Scale out blockchain with asynchronous consensus zones," in Proc. 16th USENIX Symp. Networked Syst. Des. Implementation, NSDI 2019: 95-112, 2019.

[69] Y. Liu, J. Liu, D. Li, H. Yu, Q. Wu, "Fleetchain: A secure scalable and responsive blockchain achieving optimal sharding," in Proc. International Conference on Algorithms and Architectures for Parallel Processing: 409-425, 2020.

[70] Z. Hong, S. Guo, P. Li, W. Chen, "Pyramid: A layered sharding blockchain system," in Proc. IEEE INFOCOM 2021-IEEE Conference on Computer Communications: 1-10, 2021.

[71] C. Huang et al., "RepChain: A reputation-based secure, fast, and high incentive blockchain system via sharding," IEEE Internet Things J., 8(6): 4291-4304, 2020.

[72] H. Dang, T. T. A. Dinh, D. Loghin, E. C. Chang, Q. Lin, B. C. Ooi, "Towards scaling blockchain systems via sharding," in Proc. 2019 international conference on management of data: 123–140, 2019.

[73] C. Fan, S. Ghaemi, H. Khazaei, P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," IEEE Access, 8: 126927–126950, 2020.

[74] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Bus. Rev., 2008.

[75] Y. C. Liang, "Blockchain for dynamic spectrum management," Dyn. Spectr. Manag. From Cogn. Radio to Blockchain Artif. Intell., 121–146, 2020.

[76] S. Popov, "The tangle," White Pap., 1(3): 30, 2018.

[77] R. G. Brown, J. Carlyle, I. Grigg, M. Hearn, "Corda: an introduction," R3 CEV, August, 1(15): 14, 2016.

[78] "Radix DeFi White Paper," 1–31, 2020.

[79] "The Ethereum project, the modified merkle patricia tree," 2020.

[80] O. Sury, R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC," RFC 8080 (Proposed Standard)., 2017.

[81] G. Maxwell, A. Poelstra, "Borromean ring signatures," Accessed Jun, 8: 2019, 2015.

[82] L. Lamport, "Constructing digital signatures from a one-way function," 1979.

[83] J. Reed, "Litecoin: An introduction to litecoin cryptocurrency and litecoin mining," CreateSpace Independent Publishing Platform, 2017.

[84] D. Schwartz, N. Youngs, A. Britto, "The ripple protocol consensus algorithm," Ripple Labs Inc. White Pap., 5(8): 151, 2014.

[85] "EOSIO, An Open-source Blockchain Platfom,"2018.

[86] D. Khovratovich, J. Law, "Sovrin: digital identities in the blockchain era," Github Commit by jasonalaw Oct., 17: 38-99, 2017.

[87] "LTO Network,Blockchain for Decentralized Workflows," 2019.

[88] "Holochain: Scalable Agent-Centric Distributed Computing," 2018.

[89] "Monet Network,"2018.

[90] E. Androulaki et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in Proc. the thirteenth EuroSys conference: 1-15, 2018.

[91] S. Ghimire, H. Selvaraj, "A survey on bitcoin cryptocurrency and its mining," in Proc. 2018 26th International Conference on Systems Engineering (ICSEng): 1-6, 2018.

[92] D. Ongaro, J. Ousterhout, "In search of an understandable consensus algorithm," in Proc. 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14): 305-319, 2014.

[93] C. Dannen, "Introducing Ethereum and solidity",Berkeley, 1: 159-160, 2017.

[94] A. Hafid, A. S. Hafid, M. Samih, "Scaling blockchains: A comprehensive survey," IEEE Access, 8: 125244–125262, 2020.

[95] Q. Zhou, H. Huang, Z. Zheng, J. Bian, "Solutions to scalability of blockchain: A survey," IEEE Access, 8: 16440–16455, 2020.

[96] M. H. Nasir, J. Arshad, M. M. Khan, M. Fatima, K. Salah, R. Jayaraman, "Scalable blockchains—A systematic review," Futur. Gener. Comput. Syst., 126: 136-162, 2022.

[97] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, R. P. Liu, "Survey: Sharding in blockchains," IEEE Access, 8: 14155–14181, 2020.

[98] Q. Wang, J. Yu, S. Chen, Y. Xiang, "SoK: Diving into DAG-based blockchain systems," arXiv Prepr. arXiv2012.06128, 2020.

[99] D. P. Oyinloye, J. Sen Teh, N. Jamil, M. Alawida, "Blockchain consensus: An overview of alternative protocols," Symmetry (Basel)., 13(8): 1363, 2021.

[100] J. Garzik, "Block size increase to 2MB," Bitcoin Improv. Propos., 102, 2015.

[101] B. Yu, X. Li, H. Zhao, "PoW-BC: A PoW consensus protocol based on block compression.," KSII Trans. Internet Inf. Syst., 15(4), 2021.

[102] U. Nadiya, K. Mutijarsa, C. Y. Rizqi, "Block summarization and compression in bitcoin blockchain," in Proc. 2018 International Symposium on Electronics and Smart Devices (ISESD): 1-4, 2018.

[103] S. Kim, Y. Kwon, S. Cho, "A survey of scalability solutions on blockchain," in Proc. 2018 International Conference on Information and Communication Technology Convergence (ICTC): 1204-1207, 2018.

[104] C. Decker, R. Wattenhofer, "Bitcoin transaction malleability and MtGox," in Proc. European Symposium on Research in Computer Security: 313–326, 2014.

[105] A. Back et al., "Enabling blockchain innovations with pegged sidechains," 72: 201-24 , 2014.

[106] R. A. N. Yaxuan, N. I. U. Yixin, C. Siyun, "More is less: Why multiple payment mechanism impairs individual donation," Acta Psychol. Sin., 53(4): 413, 2021.

[107] S. Bistarelli, C. Pannacci, F. Santini, "CapBAC in Hyperledger Sawtooth," in Proc. IFIP International Conference on Distributed Applications and Interoperable Systems: 152-169, 2019.

[108] "Proof of Capacity,"2018.

[109] Y. Lai, "NEM White paper," Imid 2009, (159679): 1069-1072, 2018.

[110] K. Karantias, A. Kiayias, D. Zindros, "Proof-of-burn," in Proc.

International Conference on Financial Cryptography and Data Security: 523–540, 2020.

[111] Q. Wang et al., "Security analysis on dBFT protocol of NEO," in Proc. International Conference on Financial Cryptography and Data Security: 20–31, 2020.

[112] "Steemit, A Blockchain-Based Blogging and Social Media project,"2017.

[113] "Ark Blockchain Framework,"2019.

[114] "Cardano, A Blockchain project," 2017.

[115] "Lisk Blockchain Application Platform,"2017.

[116] S. Tang, Z. Wang, J. Jiang, S. Ge, G. Tan, "Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain," Sci. Rep., 12(1): 4426, 2022.

[117] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," 2016.

[118] "Ontology, A blockchain for self-sovereign ID and DATA," 2018.

[119] P. Khahulin, I. Barinov, V. Baranov, "POA network whitepaper; technical report," 2018.

[120] "VeChainThor public Blockchain", 2019.

[121] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," 2016.

[122] Y. Sompolinsky, Y. Lewenberg, A. Zohar, "SPECTRE: a fast and scalable cryptocurrency protocol," IACR Cryptol. ePrint Arch., 2016.

[123] X. Boyen, C. Carr, T. Haines, "Graphchain: A blockchain-free scalable decentralised ledger," in Proc. the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts: 21-33, 2018.

[124] G. Srivastava, A. D. Dwivedi, R. Singh, "PHANTOM protocol as the new crypto-democracy," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics): 499–509, 2018.

[125] H. Gupta, D. Janakiram, "Cdag: A serialized blockdag for permissioned blockchain," arXiv Prepr. arXiv1910.08547, 2019.

[126] C. Li, P. Li, D. Zhou, W. Xu, F. Long, A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," arXiv Prepr. arXiv1805.03870, 2018.

[127] T. Y. Chen, W. N. Huang, P. C. Kuo, H. Chung, T. W. Chao, "DEXON: a highly scalable, decentralized DAG-based consensus algorithm," arXiv Prepr. arXiv1811.07525, 2018.

[128] X. Fu, H. Wang, P. Shi, X. Zhang, "Teegraph: A Blockchain consensus algorithm based on TEE and DAG for data sharing in IoT," J. Syst. Archit., 122: 102344, 2022.

[129] "Red-Black Merkle Tree,"2015.

[130] A. Manuskin, M. Mirkin, I. Eyal, "Ostraka: Secure blockchain scaling by node sharding," in Proc. 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW): 397-406, 2020.

[131] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in Proc. 25th {usenix} security symposium ({usenix} security 16): 279-296, 2016.

[132] S. Sen, M. J. Freedman, "Commensal cuckoo: Secure group partitioning for large-scale services," ACM SIGOPS Oper. Syst. Rev., 46(1): 33-39, 2012.

[133] I. Eyal, A. E. Gencer, E. G. Sirer, R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in Proc. 13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16): 45-59, 2016.

[134] B. S. Reddy, "securePrune: Secure block pruning in UTXO based blockchains using Accumulators," in Proc. 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS): 174–8, 2021.

[135] V. Bagaria, S. Kannan, D. Tse, G. Fanti, P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in Proc. the 2019 ACM SIGSAC Conference on Computer and Communications Security: 585-602, 2019.

[136] H. Yu, I. Nikolić, R. Hou, P. Saxena, "Ohie: Blockchain scaling made simple," in Proc. 2020 IEEE Symposium on Security and Privacy (SP): 90-105, 2020.

[137] Z. Yin et al., "Streamnet: A dag system with streaming graph computing," in Proc. the Future Technologies Conference: 499-522, 2020.

[138] J. Niu, "Eunomia: A permissionless parallel chain protocol based on logical clock," arXiv Prepr. arXiv1908.07567, 2019.

[139] M. J. Amiri, D. Agrawal, A. El Abbadi, "Caper: a cross-application permissioned blockchain," Proc. VLDB Endow., 12(11): 1385-1398, 2019.

[140] A. Gągol, D. Leśniak, D. Straszak, M. Świętek, "Aleph: Efficient atomic broadcast in asynchronous networks with byzantine nodes," in Proc. the 1st ACM Conference on Advances in Financial Technologies: 214–228, 2019.

[141] L. N. Nguyen, T. D. T. Nguyen, T. N. Dinh, M. T. Thai, "Optchain: optimal transactions placement for scalable blockchain sharding," in Proc. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS): 525-535, 2019.

[142] M. J. Amiri, D. Agrawal, A. El Abbadi, "Sharper: Sharding permissioned blockchains over network clusters," in Proc. the 2021 International Conference on Management of Data: 76-88, 2021.

[143] H. Chen, Y. Wang, "Sschain: A full sharding protocol for public blockchain without data migration overhead," Pervasive Mob. Comput., 59: 101055, 2019.

Biographies

Alemeh Matani received the B.Sc. degree from the University of Mazandaran, Iran, in 2013, and the M.Sc. degree from Kerman Graduate University of Technology, Iran, in 2015, both in Information Technology Engineering. She is currently pursuing the Ph.D. degree in Computer Engineering at the Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran. Her current research interests include blockchain, the Internet of Things (IoT) and distributed computing systems.

- Email: alemehmatani@gmail.com
- ORCID: 0009-0009-3259-736X
- Web of Science Researcher ID: JFQ-6782-2023
- Scopus Author ID: 57212064760
- Homepage: NA

Amir Sahafi received the B.Sc. degree from Shahed University of Tehran, Iran in 2005, M.Sc. and Ph.D. degrees both from Science and Research Branch of Islamic Azad University, Tehran, Iran, in 2007 and 2012, all in Computer Engineering. He is an Assistant Professor in Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran. His current research interests are Distributed and Cloud computing.

- Email: sahafi@iau.ac.ir
- ORCID: 0000-0002-6555-670X
- Web of Science Researcher ID: AAS-1208-2021
- Scopus Author ID: 24528878600
- Homepage: https://stb.iau.ir/faculty/a-sahafi

Ali Broumandnia was born in Isfahan, Iran. He received the B.Sc. degree from Isfahan university of Technology 1991, M.Sc. degree from Iran University of Science and Technology in 1995, both in Hardware Engineering and Ph.D. degree of Computer Engineering from Tehran Islamic Azad University-Science and Research Branch in 2006. From 1993 through 1995, he worked on intelligent transportation control with image processing and designed the Automatic License Plate Recognition for Tehran Control Traffic Company. He has published over 30 computer books, journal and conference papers. He is interested in Persian/Arabic character recognition and segmentation, Persian/Arabic document segmentation, medical imaging, signal and image processing, and wavelet analysis. He is reviewer of some International journals and conferences.

- Email: broumandnia@azad.ac.ir
- ORCID: 0000-0001-5145-2013
- Web of Science Researcher ID: I-6383-2018
- Scopus Author ID: 23003455800
- Homepage: https://stb.iau.ir/faculty/a-broumandnia

How to cite this paper:
A. Matani, A. Sahafi, A. Broumandnia, "A comprehensive review on blockchain scalability," J. Electr. Comput. Eng. Innovations, 12(1): 187-216, 2024.

DOI: 10.22061/jecei.2023.9975.670

URL: https://jecei.sru.ac.ir/article_2000.html