ISA Transactions 141 (2023) 30–43

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

# A survey on blockchain sharding

Xinmeng Liu, Haomeng Xie, Zheng Yan *, Xueqin Liang

*School of Cyber Engineering, Xidian University, Xi'an, China*

ARTICLE INFO

*Article history:*
Received 25 December 2022
Received in revised form 26 June 2023
Accepted 27 June 2023
Available online 1 July 2023

*Keywords:*
Blockchain
Sharding
Scalability
Parallel processing

ABSTRACT

As a promising technology, blockchain has found widespread application in numerous decentralized systems. However, the scalability problem of blockchain has drawn considerable criticism. Sharding, an effective technology, offers a solution to enhance blockchain scalability by enabling parallel validation and confirmation of transactions or new block generation. Although extensive research has been conducted on sharding, the existing literature still lacks a thorough review on its current state of arts with comprehensive analysis and evaluation. In this paper, we propose a series of evaluation criteria regarding scalability, applicability, and reliability. Additionally, we classify the cutting-edge sharding schemes based on blockchain type and sharding techniques. We then provide a comprehensive overview of these existing schemes by analyzing their respective advantages and disadvantages according to the proposed criteria. At the end of the survey, we highlight open issues and suggest future research directions based on the results of our meticulous analysis.

## 1. Introduction

Blockchain, as a fully distributed and immutable digital ledger maintained by multiple participants [1], has generated significant interest in both academia and industry. Its advanced characteristics, including decentralization, immutability, transparency and traceability, have propelled its popularity. The blockchain technology not only presents new solutions for security and trust, but also brings forth fresh opportunities and challenges for the development of decentralized applications [2,3], such as the energy industry, digital finance and supplychain management [4,5].

However, with the wide application of blockchain, existing consensus mechanisms fail to meet the demands of market development due to their low efficiency [6,7]. Currently, the issue of blockchain scalability hampers its further adoption [8]. When compared to Visa payment, which can theoretically process over 65 000 transactions per second (TPS) [9], the mainstream Bitcoin network [10] can only achieve a theoretical peak throughput of seven TPS [11]. This limitation renders it inefficient for day-to-day transactions. Thus, enhancing blockchain scalability and optimizing its throughput have emerged as pressing and highly relevant research and development areas.

Sharding represents a viable technology for enhancing blockchain throughput and reducing latency. As shown in Fig. 1, sharding was initially introduced in the realm of databases to distribute storage overhead [12]. In a blockchain network, the entire network is split into separate partitions, known as shards [13]. Unlike traditional blockchain networks, in which each node is responsible for handling all transactions, nodes in a shard only maintain a portion of the blockchain's data ledger. At present, sharding technology over the blockchain has been widely studied due to its practical demands in urgent driven by the fast development of blockchain applications.

So far, some related surveys regarding blockchain sharding have been published; however, each of these surveys has distinct focuses, as outlined in Table 1. Yu et al. [14] provided an overview of state-of-the-art sharding schemes and analyzed the theoretical upper-bound of throughput for each scheme. Wang et al. [15] primarily summarized sharding technologies of blockchain. This survey studies five key procedures in a sharding scheme and discussed major challenges associated with each procedure. Han et al. [16] decomposed the structure of sharding technology in blockchain into four foundational layers with orthogonal functionality, including the data layer, membership layer, intra-shard layer, and cross-shard layer. They focused on the coherence of system settings across layers. Liu et al. [17] presented a comprehensive framework for analyzing the security and performance of sharding schemes, considering seven phases of the process. However, their work lacks proper classification and a comprehensive review of sharding schemes. Zhou et al. [18] discussed and analyzed four representative sharding schemes (ELASTICO [19], OmniLedger [20], RapidChain [21], and Monoxide [22]). Xie et al. [23] and Kim et al. [24] conducted concise summaries of various sharding schemes, highlighting their importance, advantages, and disadvantages. Nonetheless, these two surveys lack in-depth analysis and comparison. Chauhan

* Corresponding author.
*E-mail addresses:* alexaliuxm@gmail.com (X. Liu), haomengxie@foxmail.com (H. Xie), zyan@xidian.edu.cn (Z. Yan), liangxueqin@xidian.edu.cn (X. Liang).

https://doi.org/10.1016/j.isatra.2023.06.029
0019-0578/© 2023 ISA. Published by Elsevier Ltd. All rights reserved.

**Table 1**
Comparison of our survey with other existing surveys.

| RE | Characteristic | Thorough schemes reviewed on both Pl and Pd | Classification | Criteria involved | Comparison and analysis of all reviewed schemes | Published year |
|---|---|---|---|---|---|---|
| [26] | Discuss on Ethereum sharding and detailed process | ○ | ○ | ○ | ○ | 2017 |
| [25] | Analysis the advantages and disadvantages of sharding | ○ | ◐ | ○ | ○ | 2018 |
| [23] | Analysis the particularity of sharding among scalability | ○ | ○ | ○ | ○ | 2019 |
| [15] | Analysis the consensus and atomicity of sharding | ○ | ◐ | ● | ● | 2019 |
| [14] | Classify sharding into the category of on-chain solutions | ○ | ○ | ○ | ○ | 2020 |
| [18] | Focus on the process of sharding. | ○ | ○ | ● | ● | 2020 |
| [16] | Focus on the structure of sharding. | ○ | ○ | ◐ | ● | 2021 |
| [17] | Focus on the structure of sharding. | ○ | ○ | ● | ● | 2022 |
| Our survey | Comprehensive analysis and discussion on these fine-granted classification and specific criteria. | ● | ● | ● | ● | – |

●: fully supported; ◐: partially supported; ○: not supported.
RE: reference; Pl: permissionless blockchain; Pd: permissioned blockchain.

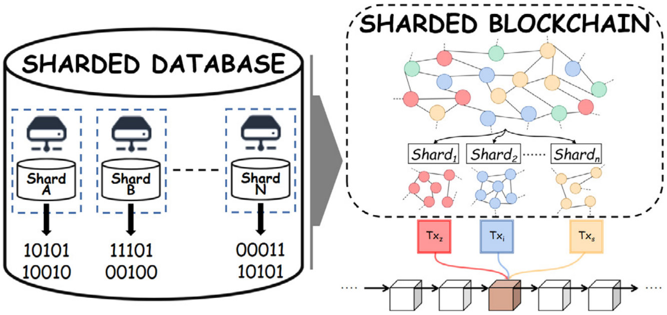

**Fig. 1.** Sharding Technology.

et al. [25] and Scherer [26] solely focused on the Ethereum sharding process. We find that although existing surveys have introduced and discussed sharding technologies, they only cover a limited number of schemes. The literature still lacks a comprehensive survey that properly classifies existing schemes and provides a thorough review using a uniform set of evaluation criteria. In addition, systematic and in-depth discussions and analyses are also missing.

In this paper, we conduct a comprehensive survey on blockchain sharding schemes. We propose a set of evaluation criteria in terms of scalability, applicability, and reliability. By employing these criteria, we review existing cutting-edge sharding schemes by classifying them based on blockchain type and sharding technique, and analyzing their strengths and weaknesses in a uniform and in-depth manner. In addition, we identify a list of open issues and propose future research directions based on the above serious review. Especially, the main contributions of this paper are summarized as below.

- We classify existing sharding schemes of blockchain according to blockchain type and sharding technique.
- We propose a series of evaluation criteria of blockchain sharding regarding scalability, applicability and reliability, based on which we conduct an in-depth review on existing sharding schemes.
- We further identify a list of open issues and highlight a number of future research directions through serious discussion and analysis on existing works.

We organize the remainder of this paper as follows. Section 2 briefly introduces the fundamental knowledge of blockchain sharding. We propose a series of evaluation criteria of sharding technology regarding scalability, applicability and reliability in Section 3. In Section 4 presents the classification of existing sharding schemes based on blockchain type and sharding technique. We analyze their advantages and disadvantages by employing the proposed evaluation criteria. Based on the literature review, we explore open issues and outline future research direction in Section 5. Finally, we conclude this paper in the last section.

## 2. Background knowledge

In this section, we first introduce the structure of blockchain. We then specify the scalability issue of blockchain and present main solutions including sharding technology in detail. In the end, we present a two-level taxonomy of sharding schemes based on blockchain type and sharding technique.

### 2.1. Blockchain structure

Blockchain is a decentralized and immutable digital ledger that operates on a peer-to-peer (P2P) network [27], where each peer maintains a complete copy of data [28]. It is first successfully applied to a distributed cryptocurrency system named Bitcoin [29,30]. The blockchain, as shown in Fig. 2, is composed of a series of blocks that continually grow over time [31]. Each block consists of a block header and a list of transactions. The block header typically contains a hash of the previous block, a timestamp, and a Merkle root [32]. By including the hash of the previous block, blocks are effectively linked together, ensuring the immutability of transactions within the blockchain. The timestamp serves as proof that the block was created at a specific moment. The Merkle root, derived through a hash function, represents the root of a binary tree structure that encodes all transactions within the block.

A blockchain can also be regarded as a replicated deterministic state machine. A state machine is an abstract mathematical model that receives inputs, then executes transitions and changes from one state to another. In a blockchain network, the notion of a replicated deterministic state machine implies that if a node starts at a particular state and replays the same sequence of transactions, it will always arrive at the same final state [33]. Thus, the state of each blockchain miner has a consistent transition when the same block is accepted. In addition, all miners in the blockchain system maintain a consistent ledger that reflects the same system status, known as the global state.

In the context of sharding technology, two types of blockchain networks are commonly discussed: permissionless blockchain and permissioned blockchain [34]. A permissionless blockchain is an open and fully decentralized network. It allows anyone to

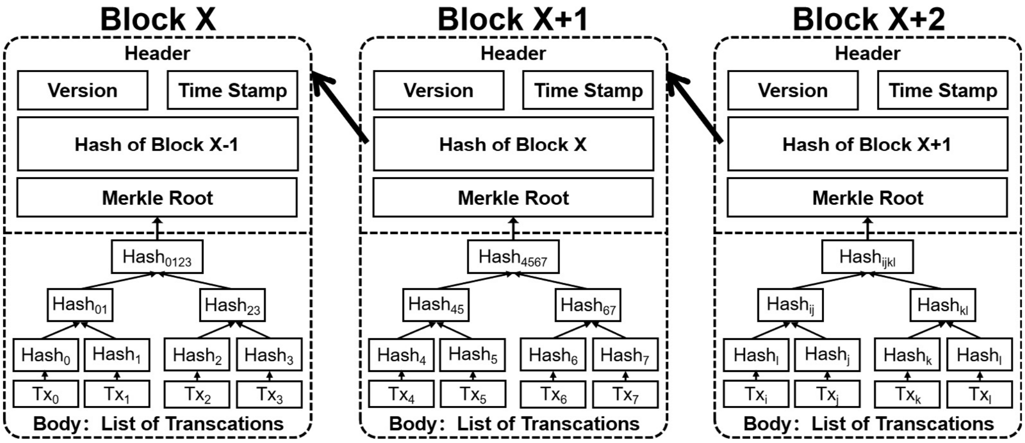

**Fig. 2.** Blockchain Structure.

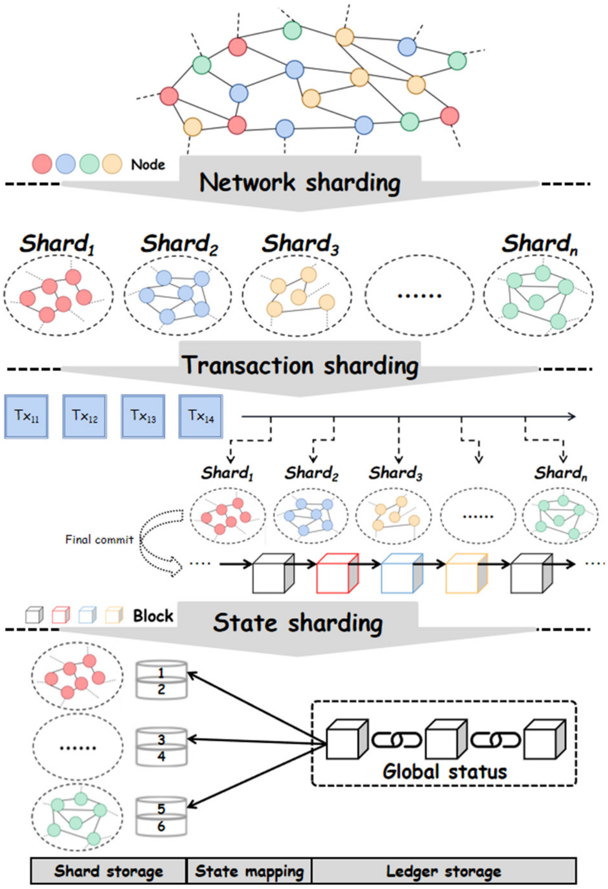

**Fig. 3.** Three Sharding Techniques.

participate in interactions and consensus validation processes. It operates on a trustless model, where no central authority is required, and all participants have equal access and control over the blockchain. In contrast, a permissioned blockchain [35] is not publicly accessible. It is a blockchain network where users can only access and perform specific operations upon obtaining permissions from the blockchain administrators [36].

In addition, the unspent transaction output (UTXO) model [37, 38] and the account model [39] are the two main transaction types in blockchain. In the UTXO model, the inputs of a new transaction are unspent outputs from previous transactions, while the outputs of a new transaction can then be used as inputs and spent in future transactions [40]. In contrast, the account model applies the current balance of each account to represent the global state [41]. In this model, the blockchain keeps track of the balance associated with each account to ensure that it has sufficient assets to cover the cost of a transaction.

### 2.2. Blockchain scalability and sharding technology

The scalability issue poses a significant challenge for the widespread adoption of blockchain technology, as many blockchain networks struggle to maintain efficiency with an increasing number of transactions [42]. It is mainly influenced by factors such as latency, throughput and communication overhead. Sharding offers a promising solution that can significantly improve the throughput, but limitedly increase the communication overhead and latency, thus relieving the scalability problem of blockchain [43].

The sharding technology originates from traditional centralized databases. It partitions the entire database into separate parts, known as shards. Similarly, in a blockchain system, sharding involves dividing the network into distinct shards, each comprising a subset of nodes. Shards handle disjoint set of transactions in parallel, thus improving the efficiency of the blockchain process.

Sharding execution normally involves four steps: data initialization, sharding establishment and setting, consensus determination, and reconfiguration. In the permissionless blockchain, nodes are required to establish identities (i.e. public key, IP address) in the process of data initialization. Then during the shard establishment and setting phase, nodes and transactions are randomly partitioned into distinct shards. Nodes within a shard can freely share their identities and establish connections with each other. Following this, nodes in the same shard conduct intra-shard consensus to agree on a set of transactions, while nodes among different shards execute cross-shard consensus to achieve a global status [44]. Finally, to maintain system security, a sharding system periodically undergoes reconfiguration to reallocate nodes into different shards.

The network status of a sharding system can be classified into three types, synchronous network (sync), partial synchronous network (psync) [45], and asynchronous network (async). In a synchronous network, the behavior and state of nodes are consistently aligned. In a partial synchronous network, although nodes may have different clocks, they can still achieve a consistent state in the end [46]. However, in an asynchronous network, nodes lack feedback about the states of other nodes, resulting in differing views on the overall state of the network. While designing a sharding scheme based on a synchronous network is straightforward, real-world scenarios often involve complex partially synchronous or asynchronous networks.

Refer to Fig. 3, there are three layers of sharding techniques: network sharding, transaction sharding, and state sharding.

- **Network Sharding (NS):** Among the different sharding techniques, network sharding serves as the most fundamental approach. It partitions all nodes in a blockchain network into different shards randomly. Normally, node allocation methods involve non-functional allocation and functional allocation. Non-functional allocation assigns nodes to shards based on their identities [22] or by employing randomness generators such as RandHound and verifiable random function (VRF) [19–21]. In contrast, functional allocation employs machine learning algorithms to provide a multifunctional approach that enhances the security and self-adaptability of the sharding system.
- **Transaction Sharding (TS):** Transaction sharding partitions transactions into different shards, in which every node remains a copy of the full blockchain and processes transactions in parallel. To avoid double-spending, transactions with the same input can be assigned to the same shard. Alternatively, transactions can also be randomly allocated to different shards according to their addresses. However, in the latter case, additional cross-chain communication is necessary to ensure that transactions with the same input but assigned to different shards are properly synchronized and prevent double-spending.
- **State Sharding (SS):** State sharding splits the global state of a blockchain network into different shards. Unlike transaction sharding, each node in state sharding only stores a portion of the complete ledger, which significantly reduces its storage overhead. However, it may introduce additional cross-shard communication overhead during the phase of data initialization and sharding establishment. In addition, it requires extra backups to maintain the complete global state of the blockchain to prevent corruption within a shard.

## 3. Evaluation criteria on sharding

Evaluation criteria play a crucial role in the assessment of blockchain sharding schemes, as they facilitate comparison and identification of strengths and weaknesses of different schemes. Based on the evaluation criteria, we can conduct a thorough and rigorous analysis on different sharding schemes accordingly, enabling the identification of open issues and future research directions. In this section, we present a list of evaluation criteria with regard to scalability, applicability, and reliability in order to evaluate the performance of sharding schemes discussed in Section 4 and highlight open research issues outlined in Section 5. The taxonomy of the evaluation criteria is depicted in Fig. 4.

### 3.1. Criteria on scalability

Scalability refers to the ability of a sharding scheme to perform well even with an expanding workload. A scalable blockchain system can maintain or even enhance its performance even when faced with a large number of transactions. The scalability of a sharding scheme can be evaluated based on three key aspects: latency, throughput and communication overhead.

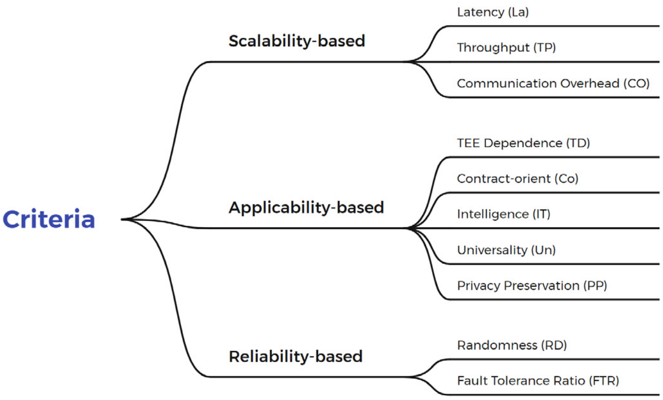

**Fig. 4.** Evaluation Criteria.

#### 3.1.1. Latency (La):

The latency of a transaction refers to the duration between submitting a transaction to the blockchain and the validation of a block that contains the transaction [47]. Low latency is crucial for real-time applications, as it directly affects the response time and processing speed. Normally, the shard size (i.e. the number of nodes in a shard) typically has a significant impact on the transaction latency.

#### 3.1.2. Throughput (TP):

Throughput serves as a crucial quality indicator for evaluating the scalability of a sharding scheme [48]. Increasing the throughput is essential for enabling a blockchain network to process a large volume of transactions, which, ultimately, leads to a reduction in the overall cost of managing the network. Normally, throughput is measured by the number of transactions a blockchain system can process within a given time period.

#### 3.1.3. Communication overhead (CO):

The communication overhead of a sharding scheme refers to the data transmitted among shards for validating intra-shard and cross-shard transactions. It directly impacts blockchain consensus efficiency and, consequently, affects the throughput. For ease of comparison and analysis, communication complexity is employed as a metric to quantify the communication overhead. It is influenced by various factors, including the number of nodes in the network ($N$), the number of nodes in a shard ($m$), and the number of shards in the network ($n$).

### 3.2. Criteria on applicability

The real value of a sharding scheme lies in its applicability, which refers to its ability to adapt to diverse scenarios. We propose the following criteria to evaluate the applicability of a sharding scheme.

#### 3.2.1. TEE dependence (TD):

TEE dependence refers that a sharding scheme is constructed based on a trusted execution environment (TEE) [49,50]. The dependence on TEE creates a reliance on the underlying hardware and software platforms, thereby limiting the generality of a sharding scheme [51]. TEEs provide capabilities such as data encryption, secure computations, and access control enforcement [52]. While TEEs offer a certain level of flexibility and security, their usage may impose constraints on the overall adaptability and portability of the sharding scheme.

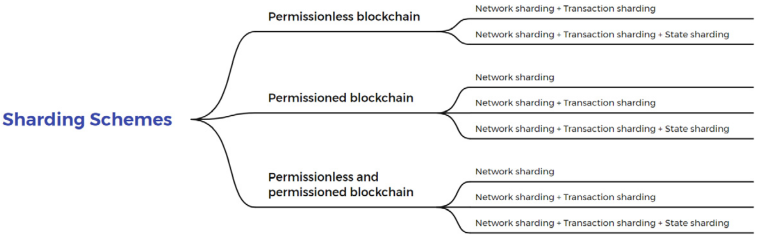
Sharding Schemes
Permissionless blockchain
Network sharding + Transaction sharding
Network sharding + Transaction sharding + State sharding
Permissioned blockchain
Network sharding
Network sharding + Transaction sharding
Network sharding + Transaction sharding + State sharding
Permissionless and permissioned blockchain
Network sharding
Network sharding + Transaction sharding
Network sharding + Transaction sharding + State sharding

**Fig. 5.** The classification of sharding schemes.

#### 3.2.2. *Contract-orientation (Co):*

Contract-orientation refers that a sharding scheme is compatible with smart contracts. It enables a set of conditional digital commitments to be executed correctly according to previous agreement [53,54]. Meanwhile, a contract-orientated sharding scheme can reduce the dependence on a trusted third party, thereby improving security. For example, smart contracts can be used to automatically validate transactions and enforce rules such as transaction fees, transaction size limits, and data privacy requirements [55].

#### 3.2.3. *Intelligence (IN):*

During the reconfiguration of a sharding system, a large number of parameters are required to be adjusted according to different application scenarios. The intelligence of a sharding scheme refers to its capability to automatically optimize these parameters using a learning method (e.g., machine learning) [56–58]. By employing intelligent parameter adjustment, a sharding scheme can enhance its efficiency, scalability and adaptability.

#### 3.2.4. *Universality (Un):*

Universality refers to the capacity of a sharding scheme to adapt to diverse application scenarios. It allows for seamless integration into different blockchain systems or networks, and provides a versatile solution that accommodates different requirements and environments. A universal sharding scheme is highly flexible, improving its applicability and reducing maintenance costs.

#### 3.2.5. *Privacy preservation (PP):*

Due to the openness of blockchain, the data and identity of users recorded on the blockchain are consequently open to the public. Thus, privacy preservation becomes crucial to avoid the leakage of sensitive user data [59,60]. A sharding scheme should apply effective methods to prevent malicious nodes from colluding across shards to compromise the privacy of users' data and identities.

### 3.3. *Criteria on reliability*

A sharding scheme is expected to be executed correctly even in some malicious environments. Therefore, we propose the following criteria to evaluate the reliability of a sharding scheme.

#### 3.3.1. *Randomness (RD):*

Randomness plays a crucial role in ensuring the fair distribution of malicious nodes across different shards, thus reducing the probability of an adversary gaining control over a single shard. Permitting nodes to choose a shard that they want to join is insecure, since an adversary can master all malicious nodes to join and control a shard, which breaks blockchain robustness.

#### 3.3.2. *Fault tolerance (FT):*

Fault tolerance of a sharding scheme refers to its capability to preserve the accurate state and functionality of the blockchain network in the presence of byzantine nodes. It is evaluated by fault tolerance ratio (FTR), which is measured by (N - k) / N, where $N$ is the total number of nodes in a shard or a global network, and k is the maximum number of byzantine nodes when the network can operate normally. In a blockchain network, byzantine nodes always broadcast inconsistent information to hinder the reach of consensus [61]. Therefore, a robust blockchain network always requires a high fault tolerance ratio [62]. We measure fault tolerance at both intra-shard and global network levels to ensure the overall robustness and security of a sharding scheme.

## 4. Review on sharding schemes

In this section, we present a comprehensive review of state-of-the-art sharding schemes published in the past decade, focusing on blockchain scalability, sharding, and scaling blockchain from the following databases: IEEE Explorer Digital Library, ACM Digital Library, Elsevier ScienceDirect, Springer, Engineering Village, and Web of Science. We analyze and evaluate the scalability, applicability and reliability of each scheme based on the above proposed evaluation criteria. Finally, we sum up and compare all reviewed sharding schemes with regard to the criteria in Table 2.

As shown in Fig. 5, we classify the sharding schemes based on supported blockchain type, i.e., permissionless blockchain (Pl), permissioned blockchain (Pd), or both, and then according to sharding technique, i.e., network sharding, transaction sharding, or state sharding. Since we found that all below reviewed schemes apply network sharding, thus we classify applied sharding techniques into three classes: using network sharding, using network sharding and transaction sharding, and using network sharding, transaction sharding and state sharding.

### 4.1. *Sharding schemes in permissionless blockchains*

We begin by providing an overview and analysis of existing sharding schemes for permissionless blockchains. As there are no sharding schemes based solely on network sharding, we review schemes based on network sharding and transaction sharding, as well as network sharding, transaction sharding and state sharding.

#### 4.1.1. *Network sharding and transaction sharding*

The following schemes implement both network sharding and transaction sharding in the permissionless blockchain network.

***ELASTICO:*** Luu et al. [19] first introduced the sharding technology in the permissionless blockchain, specifically within a partial synchronous network environment. In ELASTICO, all validators have similar computation and network resources. As shown in Fig. 6, it first executes the network sharding to partition nodes

**Table 2**
Summary on sharding schemes in blockchain.

| Ref | BT | *Sharding* | | | TT | Ns | ISCM | CSCM | *Scalability* | | | | *APPLicable* | | | | | *Reliability* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | NS | TS | SS | | | | | La | TP | CO | | TD | Co | IN | Un | PP | RD | FT |
| | | | | | | | | | | | ISCO | CSCO | | | | | | | |
| [19] | Pl | ✓ | ✓ | × | UTXO | Psync | PBFT | PBFT | $\mathcal{O}(1)$ | $\mathcal{O}(\frac{n}{\log n})$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(n)$ | × | × | × | × | × | ✓ | 33%/25% |
| [63] | Pl | ✓ | ✓ | × | Account | Async | PBFT | | – | $\mathcal{O}(n)$ | $\mathcal{O}(m^2)$ | – | × | ✓ | × | × | × | × | 33%/33% |
| [64] | Pl | ✓ | ✓ | × | Account | Sync | BFT | – | – | – | $\mathcal{O}(m^2)$ | – | × | ✓ | × | ✓ | × | ✓ | 33%/33% |
| [65] | Pl | ✓ | ✓ | × | UTXO | Psync | NFR | NFR | $\mathcal{O}(n \log n)$ | $\mathcal{O}(\log n)$ | – | – | × | × | ✓ | ✓ | × | × | – |
| [66] | Pl | ✓ | ✓ | × | UTXO | Psync | CSBFT | Atomix | $\mathcal{O}(m)$ | $\mathcal{O}(m)$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(n)$ | × | × | × | ✓ | × | ✓ | 33%/33% |
| [67] | Pl | ✓ | ✓ | ✓ | Account | Sync | NFR | NFR | $\mathcal{O}(\frac{1}{n})$ | $\mathcal{O}(N)$ | – | – | × | ✓ | ✓ | ✓ | × | ✓ | – |
| [21] | Pl | ✓ | ✓ | ✓ | UTXO | Sync | RSC | FCSV | $\mathcal{O}(1)$ | $\mathcal{O}(N)$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(m^2 + m \log n)$ | × | × | × | × | × | ✓ | 50%/33% |
| [22] | Pl | ✓ | ✓ | ✓ | Account | Async | PoW | Eventual Atomicity | $\mathcal{O}(\log n)$ | $\mathcal{O}(n)$ | $\mathcal{O}(m)$ | $\mathcal{O}(m + n)$ | × | × | × | × | × | × | 50%/50% |
| [68] | Pl | ✓ | ✓ | ✓ | UTXO | Psync | PoW | PoW | $\mathcal{O}(N)$ | $\mathcal{O}(n)$ | $\mathcal{O}(m^2)$ | – | × | × | ✓ | × | × | ✓ | 50%/50% |
| [69] | Pl | ✓ | ✓ | ✓ | Account | Async | Casper FFG | PoS | – | – | $\mathcal{O}(m^2)$ | $\mathcal{O}(n)$ | × | × | × | × | × | ✓ | 33%/33% |
| [70] | Pl | ✓ | ✓ | ✓ | Account | Psync | – | PBFT | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ | – | $\mathcal{O}(n^2 + m^2)$ | × | ✓ | × | ✓ | × | ✓ | – |
| [71] | Pl | ✓ | ✓ | ✓ | NFR | Sync | NFR | NFR | $\mathcal{O}(1)$ | $\mathcal{O}(N)$ | – | $\mathcal{O}(\log N)$ | × | × | × | ✓ | × | ✓ | – |
| [72] | Pd | ✓ | × | × | UTXO | Async | PBFT | PBFT | – | $\mathcal{O}(N)$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(N)$ | × | × | × | × | × | ✓ | 33%/33% |
| [73] | Pd | ✓ | ✓ | × | UTXO | Async | 2PC-based consensus | – | $\mathcal{O}(1)$ | $\mathcal{O}(N)$ | $\mathcal{O}(m^2)$ | – | × | × | × | × | × | ✓ | – |
| [74] | Pd | ✓ | ✓ | × | Account | Psync | AHL | 2PL, 2PC | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ | $\mathcal{O}(m^2 n^2)$ | $\mathcal{O}(mn \log mn)$ | ✓ | ✓ | × | × | × | ✓ | 33%/33% |
| [75] | Pd | ✓ | ✓ | × | Account | Sync | PBFT | PBFT | $\mathcal{O}(\frac{Tx}{n})$ | – | $\mathcal{O}(mn)$ | $\mathcal{O}(mn)$ | × | × | × | ✓ | × | ✓ | 33%/33% |
| [76] | Pd | ✓ | ✓ | ✓ | Account | Async | CasperCBC | CasperCBC | – | $\mathcal{O}(N)$ | – | $\mathcal{O}(n^2)$ | × | ✓ | × | × | × | × | 33%/33% |
| [77] | Pd | ✓ | ✓ | ✓ | Account | Async | S-BAC | – | $\mathcal{O}(1)$ | $\mathcal{O}(\frac{n}{m})$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(n^2 + m)$ | × | ✓ | × | × | ✓ | × | 33%/25% |
| [78] | Pd | ✓ | ✓ | ✓ | UTXO | Sync | SCTP | ACC | – | – | $\mathcal{O}(m^2)$ | $\mathcal{O}(n^2 + m)$ | × | × | × | ✓ | ✓ | × | – |
| [79] | Pd | ✓ | ✓ | ✓ | Account | Async | PBFT | multi-Paxos | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ | $\mathcal{O}(m^2)$ | $\mathcal{O}(n^2)$ | × | × | × | ✓ | × | ✓ | – |
| [80] | Pd | ✓ | ✓ | ✓ | Account | Psync | NFR | NFR | $\mathcal{O}(n)$ | $\mathcal{O}(n)$ | – | – | × | ✓ | × | ✓ | × | × | 33%/33% |
| [81] | Pl&Pd | ✓ | × | × | Account | Sync | PBFT | – | – | – | $\mathcal{O}(m^2)$ | – | × | × | × | ✓ | × | ✓ | 33%/33% |
| [82] | Pl&Pd | ✓ | ✓ | × | UTXO | Psync | FBFT | RSTP | – | – | $\mathcal{O}(m^2)$ | $\mathcal{O}(n^2 + m)$ | × | × | × | ✓ | × | × | 33%/33% |
| [83] | Pl&Pd | ✓ | ✓ | × | UTXO | Sync | NFR | – | $\mathcal{O}(tmn)$ | $\mathcal{O}(N)$ | – | – | × | × | × | ✓ | × | × | – |
| [84] | Pl&Pd | ✓ | ✓ | × | Account | Psync | PBFT | – | $\mathcal{O}(1)$ | $\mathcal{O}(N), N < 1000$<br>$\mathcal{O}(1), N > 1000$ | $\mathcal{O}(m^2)$ | – | × | × | ✓ | ✓ | × | ✓ | 33%/33% |
| [20] | Pl&Pd | ✓ | ✓ | ✓ | UTXO | Psync | ByzCoinX | Atomix | $\mathcal{O}(N)$ | $\mathcal{O}(n)$ | $\mathcal{O}(\log m)$ | $\mathcal{O}(n)$ | × | × | × | ✓ | × | ✓ | 33%/25% |
| [85] | Pl&Pd | ✓ | ✓ | ✓ | UTXO | – | – | – | $\mathcal{O}(n)$ | $\mathcal{O}(\log n)$ | – | – | × | × | × | ✓ | × | × | – |
| [86] | Pl&Pd | ✓ | ✓ | ✓ | Account | NFR | NFR | NFR | – | – | – | – | × | × | × | ✓ | × | × | – |

✓: satisfied; ×: unsatisfied; –: not given; NFR: no fixed requirement.
N: the number of nodes in the network; m: the size of a shard; n: the number of shards; Tx: the number of transactions; t: running time.
Reference (Ref), Blockchain Type (BT), Permissionless (Pl), Permissioned (Pd), Networking Sharding (NS), Transaction Sharding (TS), State Sharding (SS), Transaction Type (TT), Network Status (Ns), Intra-shard Consensus Mechanism (IS-CM), Cross-shard Consensus Mechanism (CSCM), Latency (La), Throughput (TP), Communication Overhead (CO), Intra-shard Communication Overhead (ISCO), Cross-shard Communication Overhead (CSCO), Dependency (De), Contract-orient (Co), Intelligence (IN), Universality (Un), Privacy Preservation (PP), Randomness (RD), Fault Tolerance (FT): Intra-shard FTR/Global FTR.

into different committees (shards) based on a proof-of-work solution [87] and identity establishment mechanism (PoW-ID). Intra-shard Byzantine fault tolerance (BFT) consensus [88] is then run by each committee with 33% FTR to agree on a block containing UTXO-based transactions. Finally, a directory committee collects blocks from each committee and verifies signatures to generate a final block and reach a consistent global status. The latency always keeps constant and the throughput as well as the cross-shard communication overhead in ELASTICO increase almost linearly with the number of shards. Meanwhile, its intra-shard communication overhead increases quadratically with the size of shards. To prevent a nonce for locating shards from being calculated in advance and submitted in the next epoch, ELASTICO introduces an EpochRandomness variable as a random seed for an epoch. Furthermore, odes and processors in the network are assigned to different committees corresponding to their identities, thus **RD** is guaranteed. However, **Co**, **IN**, **Un** and **PP** are overlooked.

***Zilliqa:*** Zilliqa [63] is an innovative cryptocurrency platform that utilizes a scalable collective signing scheme (CoSi) [89] to support asynchronous permissionless blockchain. Account-based transactions are validated through the EC-Schnorr-based practical byzantine fault tolerance (PBFT) consensus [90–92], which can tolerate up to 33% byzantine nodes in the network. Its throughput increases linearly with the number of shards. Meanwhile, its communication overhead increases quadratically as the size of shard increases. Zilliqa supports **Co** by using a formally verifiable language called Scilla, which is sharding-friendly and suitable for running massive parallel and complex arithmetic computations. However, it is worth noting that Zilliqa involves a two-phase of PoW-based and address-based node allocation that cannot reflect the criterion of **RD**. **IN**, **Un** and **PP** were not mentioned.

***Poster:*** Lee et al. proposed a dynamic shard management blockchain protocol named Poster [64], which is based on Proof of Stake (PoS) and operates in a synchronous network. This protocol effectively solves the unfair allocation problem of nodes and transactions in sharding. Poster is a type of flexible shard establishment protocol based on account-based transactions. The shard block is accepted following the BFT intra-shard consensus, thus tolerating 33% byzantine nodes. Its intra-shard communication

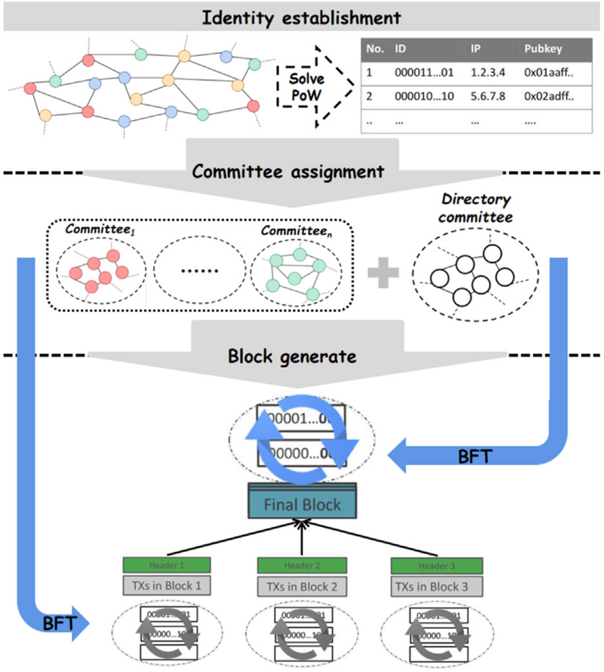

**Fig. 6.** ELASTICO.

overhead increases quadratically with the size of shard. In addition, it can be applied to other sharding schemes, thus supporting **Un**. At the same time, Poster assigns validators and distributes transactions into shards via smart contracts. The protocol applies the Byzantine Fault Tolerance Delegated Proof of Stack (BFT-DPoS) algorithm to select block producers, thus guaranteeing **Co** and **RD**. But the protocol fails to deal with **IN** and **PP**.

***Optchain:*** In order to decrease the overhead of cross-shard transactions, Hguyen et al. [65] proposed a lightweight and real-time UTXO-based transaction allocation strategy in a partial synchronous network, named OptChain. Its latency increases log-linearly with the number of shards, and the throughput increases logarithmically with the number of shards. Transactions and nodes are abstracted and built as an online directed acyclic graph (DAG) [93] to construct a Transactions as Nodes (TaN) network, which optimizes the overhead of cross-shard transactions. Given that, Optchain satisfies the criterion of **IN** and supports **Un** as it does not interfere with the core consensus protocol and can be deployed in existing wallet software. However, it ignores to support **PP**, **Co** and **RD**.

***Repchain:*** Huang et al. [66] proposed Repchain, the first reputation-based double-chain system with incentive-using sharding [94]. As shown in Fig. 7, all nodes in RepChain are assigned into different shards randomly, and a shard leader is selected in each shard first. Each shard generates a reputation chain based on transaction records via intra-shard collective signing BFT (CSBFT) consensus and Atomix consensus [21,95] to tolerate 33% byzantine nodes, based on which a transaction chain is generated via Raft consensus [96,97]. After that, each shard concludes the transaction chain and the reputation chain in a state block. At the end of each epoch, Repchain can execute state synchronization and update to reach a consistent global state based on state blocks generated by shards. Its latency and throughput increase linearly with the size of shard. The intra-shard communication overhead is correlated with the square of the shard size, while the cross-shard communication overhead increases linearly with the number of shards. The reputation mechanism in Repchain can be applied into other sharding systems to improve overall security, thus satisfying the criterion of **Un**. Moreover, Repchain provides a secure and distributed bias-resistant random generation protocol [98], which is similar to that in OmniLedger and RapidChain. Therefore, it guarantees **RD**. However, it cannot achieve the goals of **IN**, **PP** and **Co**.

### 4.1.2. Network sharding, transaction sharding and state sharding

The following schemes implement all network sharding, transaction sharding and state sharding in the permissionless blockchain network.

***Sharding in Open Blockchains with Smart Contracts:*** Tao et al. [67] proposed a distributed dynamic sharding contracts-based system that aims to minimize cross-shard communication with an isolation validation method and significantly increase the throughput of blockchain systems. The latency of this scheme decreases linearly as the number of shards increases, while its throughput is in proportion to the number of nodes in the network. To further optimize the system, they designed an intra-shard merging algorithm to dynamically merge small shards to form a large shard, in which miners exchange and modify merging choices until a mixed Nash equilibrium has been achieved in an epoch. Therefore, this system supports **IN** to some extent. In addition, It applies a random selection algorithm to decide the allocation of miners and transactions to each shard, which ensures **RD**. It can be applied into multiple scenarios without specific requirements on consensus mechanisms, thus satisfying **Un**. However, there is no discussion on **PP**.

***RapidChain:*** Zamani et al. [21] proposed a slowly-adaptive Byzantine adversary resilience sharding blockchain scheme called RapidChain based on the Cuckoo rule [99]. It applied rapid shard consensus (RSC) with 33% FTR to validate intra-shard transactions. Deployed on a synchronous network with UTXO-based transactions, RapidChain reduced the delay that keeps constant as the network size increases. The throughput increases linearly with the total number of nodes. Meanwhile, the intra-shard communication overhead grows quadratically as the network size increases, while the complexity of cross-shard communication overhead is $\mathcal{O}(m^2 + m\log n)$. In RapidChain, each shard internally runs a distributed random generation (DRG) protocol to generate an unbiased random value to build a reference committee to assign nodes randomly. Therefore, it satisfies the criterion of **RD**. Unfortunately, RapidChain does not support **IN**, **Un**, **PP** and **Co**.

***Monoxide:*** Wang et al. proposed Monoxide, a concurrent multichain system, to address the issue of heavy cross-shard communication overhead in an asynchronous network [22]. As shown in Fig. 8, it first partitions network nodes and transactions into different asynchronous consensus zones (i.e. shards). It applied PoW consensus mechanism with 50% FTR to validate intra-shard transactions. In the process of cross-shard communication, Eventual Atomicity consensus verifies operations and validates account-based transactions in each zone to ensure atomicity. In addition, they proposed a new network sharding scheme called Chu-ko-nu Mining to support a miner to validate multiple blocks in different zones simultaneously. The latency and throughput in Monoxide increase logarithmically and linearly with the number of shards respectively. Its intra-shard communication overhead is proportional to the number of shards, while the complexity of its cross-shard communication is $\mathcal{O}(m+n)$. Unfortunately, Monoxide divides nodes into subsets deterministically without considering **RD**. In addition, **Co**, **Un**, **PP** and **IN** were not explored.

***SSChain:*** The reconfiguration process in a sharding scheme effectively prevents a slowly-adaptive adversary, nevertheless resulting in heavy bandwidth consumption and time overhead. To solve this problem, Chen et al. [68] proposed SSChain, which involves a two-layer structure without periodic network reshuffling

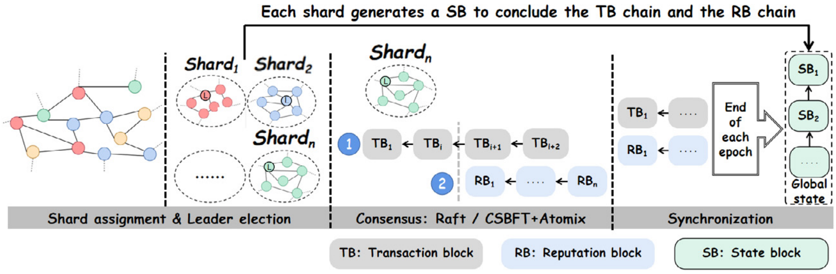

**Fig. 7.** Repchain.

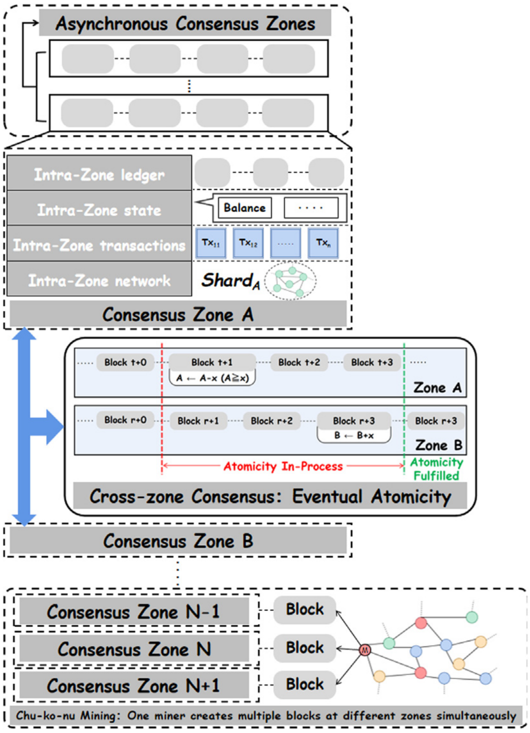

**Fig. 8.** Monoxide.

and splits cross-shard transactions into intra-shard transactions. It is a byzantine-resilient blockchain for UTXO transactions, in which nodes can join one or more shards based on the PoW-ID mechanism. The latency and the throughput of SSChain are proportional to the number of shards and the size of shard respectively. Meanwhile, its intra-shard communication overhead grows quadratically with the size of shard. It applied PoW consensus mechanism to validate intra-shard transactions and maintain the global state, which can tolerate 50% byzantine nodes. SSChain adopts a market incentive mechanism to dynamically adjust computation power distribution. Thus, SSChain supports **RD** and **IN**, but overlooks other criteria such as **PP**, **Un** and **Co**.

***Ethereum 2.0:*** Ethereum 2.0 (Eth2.0) [69] is Buterin's proposal based on Ethereum 1.0 (Eth1.0), featuring a Beacon chain and 64 shard chains. The Beacon chain designates a validator committee for each shard to coordinate all validators in shards. Each shard uses a hybrid PoS consensus mechanism [100] for intra-shard consensus, named Casper Friendly Finality Gadget (Casper FFG), which can tolerate 33% byzantine nodes. Committees are frequently altered and selected with multi-stage random number generation (RNG) [101] and verifiable delay function RANDAO [102], thus Ethereum 2.0 meets the criterion of **RD**. In terms of communication overhead, the intra-shard communication overhead is correlated with the square of the shard size, while cross-shard communication overhead increases linearly with the number of shards. So far, Ethereum 2.0 can only support simple account-based intra-shard transactions, thus it fails to meet **Co**, **IN** and **Un**. In addition, **PP** has not yet be addressed.

***Pyramid:*** Hong et al. [70] proposed the first layered sharding blockchain system with layered sharding consensus, called Pyramid, where shards can overlap with each other and a node can reside in multiple shards. Pyramid employs a novel hierarchical sharding consensus that enables partial shards to store the global state of the blockchain, validating and committing cross-shard transactions in one round. Its complexity of cross-shard communication is $\mathcal{O}(m^2+n^2)$. Meanwhile, its latency and throughput grow linearly with the number of nodes in the network. Pyramid pays attention to layered sharding architecture and consensus without specific blockchain system and scenario restrictions, thus satisfying **Un**. Meanwhile, it supports smart contract, achieving the goal of **Co**. Furthermore, **RD** can be satisfied as it applies the PoW-ID mechanism to randomly allocate nodes into shards. Unfortunately, **IN** and **PP** were not considered.

***Free2Shard:*** Rana et al. [71] proposed Free2Shard, a reputation-based dynamic self-allocation policy for a synchronous network. It introduces a multi-consensus architecture, in which shards can adopt different consensus mechanisms. Its latency remains constant, while its throughput and cross-shard communication overhead increase linearly and logarithmically with the number of nodes in the network respectively. Free2Shard applies a dynamic self-allocation (DSA) algorithm, with which nodes are allocated randomly. Therefore, it meets the criterion of **RD**. Meanwhile, Free2Shard is adaptable to multiple scenarios and any type of transaction, thus supporting **Un**. However, Free2Shard neglects **IN**, **PP** and **Co**.

### *4.2. Sharding schemes in permissioned blockchains*

In this part, we review the existing sharding schemes for the permissioned blockchains.

#### *4.2.1. Network sharding*

The following scheme only implements network sharding in the permissioned blockchain network.

***A scale-out blockchain for value transfer with spontaneous sharding:*** Ren et al. [72] proposed a value transfer sharding system in the asynchronous network based on the Value-Transfer Ledgers (VTL) without sacrificing reliability and decentralization. The system employs PBFT intra-shard consensus, enabling it to withstand up to 33% byzantine nodes. Moreover, its throughput and cross-shard communication overhead grow linearly as the number of network nodes increases, while the intra-shard communication overhead is proportional to the square of the shard size. This system can be only applied to permissionless blockchains using the PBFT consensus mechanism. Therefore, it does not satisfy **Un**. In addition, **Co**, **IN**, **PP**, and **RD** were not considered.

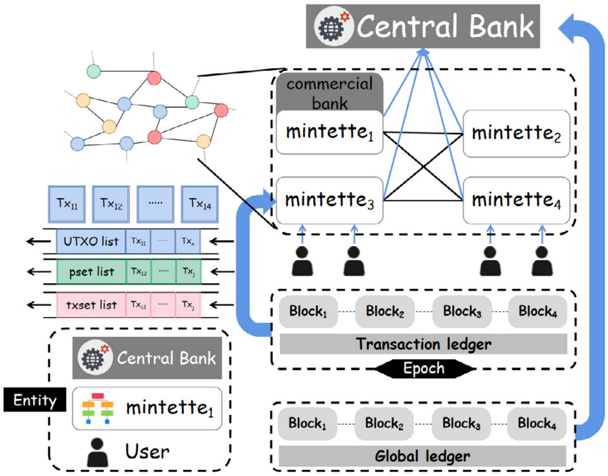

**Fig. 9.** RSCoin.

### 4.2.2. Network sharding and transaction sharding

The following schemes employ both network sharding and transaction sharding in the permissioned blockchain network.

***RSCoin:*** Danezis and Meiklejohn introduced an auditable dual-layered cryptocurrency system called RSCoin [73] to help banks control the trade of digital currency in the asynchronous network. As shown in Fig. 9, central banks manage the money supply and delegate the validation authority to other institutions, known as mintettes or shards. In each epoch, every mintette generates a transaction chain. After that, the central bank updates a global state at the end of several epochs. Its latency remains constant but its throughput increases linearly with the total number of nodes, while its intra-shard communication overhead grows quadratically with the shard size. In addition, it cannot support **Un**, because it can only be applied to a bank system. Unfortunately, it ignores consideration of **Co**, **IN**, **RD**, and **PP**.

***Scaling blockchain systems via sharding:*** Dang et al. [74] proposed a sharding scheme for a partial synchronous environment based on the Intel Software Guard Extensions (SGX) [103]. The scheme uses a BFT reference committee, two-phase commit (2PC), and two-phase locking (2PL) protocol [104] to validate account-based transactions, which can tolerate up to 50% byzantine nodes. The latency and throughput increase linearly as the network nodes increase. However, its intra-shard communication overhead grows quadratically with the size and number of shards respectively, and the cross-shard communication complexity comes up to $\mathcal{O}(mn \cdot logmn)$. This scheme applies a trusted randomness beacon inside SGX to generate unbiased random values, enabling random node allocation. Therefore, it satisfies **RD** and **TD**. Moreover, it supports **Co** with chaincodes. Besides, this scheme fails to deal with **Un**, **IN** and **PP**.

***Pruneable sharding-based blockchain protocol:*** Feng et al. [75] presented a pruneable sharding-based blockchain (PSRB) consensus on the basis of the Rollerchain system [105], which can tolerate up to 50% byzantine nodes. It avoids a serious cubical dilatation problem by storing only the own transactions nodes processed and the blockheader chain [106] of other transactions and some blocks including the genesis block and the last blocks for avoiding forking in the local storage in each node. The latency is proportional to the number of transactions but inversely proportional to the number of shards. Moreover, its intra-shard and cross-shard communication overhead increase linearly as the size and the number of shards increase respectively. With the PoW-ID mechanism, nodes and transactions are randomly and evenly assigned to each shard, thus fulfilling **RD**. This scheme can be used in multiple PBFT-based sharding scenarios, somewhat satisfying **Un**. **IN** and **Co** were not discussed. It cannot support **PP**, either.

### 4.2.3. Network sharding, transaction sharding and state sharding

The following schemes employ network sharding, transaction sharding and state sharding as a whole in the permissioned blockchain network.

***RChain:*** Greg et al. [76] proposed a tree-based sharding architecture called RChain in an asynchronous network with account-based transactions. It achieves consistency via the “Correct-by-Construction” family of Casper (CasperCBC) consensus mechanism, by which it can tolerate up to 33% byzantine nodes. Its throughput grows linearly with the number of nodes, but cross-shard communication overhead increases quadratically with the number of shards. In addition, RChain supports smart contracts with RhoLang language, thus satisfying the criterion of **Co**. However, it overlooks the criteria of **PP**, **RD**, **Un** and **IN**.

***Chainspace:*** Mustafa et al. [77] proposed a cryptocurrency framework, which can efficiently process account-based transactions in an asynchronous environment. Chainspace proposes a distributed consistency commit consensus called Sharded Byzantine Atomic Commit (S-BAC) to for sharding generic smart contract transactions across multiple byzantine nodes, and correctly coordinating those nodes to ensure security. It can tolerate up to 33% malicious nodes, with constant latency and linearly growing throughput as the number of shards increases. Intra-shard communication overhead increases quadratically with shard size, while cross-shard communication overhead is proportional to the square of shard size and shard number. It supports users to construct privacy-friendly smart contracts via ManageContracts. In addition, it integrates a security commitment and Succinct Argument of Knowledge (SNARK) to protect privacy. Thus, it meets the criteria of **Co** and **PP**. Node allocation depends on the code of smart contracts, which does not meet the criterion of **RD**. Unfortunately, **Un** and **IN** were ignored.

***Channels:*** Androulaki et al. [78] proposed Channel, a state sharding scheme with less computation overhead in a synchronous environment with UTXO-based transactions. It applied single-channel transaction protocol (SCTP) to validate intra-shard transactions and atomic cross-channel (ACC) consensus to achieve a consistent global state. Its intra-shard communication overhead grows quadratically with the shard size, while its cross-shard communication overhead is in proportion to the square of shard size and shard number. Meanwhile, it ensures **PP** through a confidential channel protocol and group key agreement. Besides, it can be applied into multiple permissioned blockchain systems under different trust assumptions, which makes it satisfy **Un**. However, node allocation is not considered, making it difficult to evaluate **RD**. In addition, **Co** and **IN** were not mentioned in this work.

***SharPer:*** Amiri et al. [79] proposed an efficient sharding scheme in an asynchronous network environment with crash-only fault-tolerant (CFT) nodes [107]. It applied PBFT and multi-Paxos [108] consensus to validate account-based transactions, by which it can tolerate up to 50% byzantine nodes. The latency and throughput increase linearly as the number of network nodes increases. Meanwhile, its intra-shard and cross-shard communication overhead grow quadratically with the shard size and shard number respectively. SharPer can be applied into other permissioned blockchains, thus satisfying **Un**. However, it does not meet the criterion of **RD**, as nodes are assigned to different shards based on their geographical distribution. In addition, it does not explore **IN**, **PP** and **Co**.

***Meepo:*** Zheng et al. [80] presented Meepo, which enhances cross-shard efficiency of account-based transactions by utilizing a merging sending data strategy and an efficient cross-shard validation protocol. Meepo uses proof of authority (PoA) as the validator select protocol. Meanwhile, its latency and throughput increase linearly with the number of nodes in the network. It applies smart contracts to address the problem of multi-state dependency when merging sent data, thus achieving **Co**. Meepo supports multiple types of consensus mechanisms, thus it supports the **Un**. However, **RD**, **PP** and **IN** were not mentioned in this work.

### *4.3. Sharding schemes in permissionless blockchains and permissioned blockchains*

In this part, we review the existing sharding schemes for both permissionless and permissioned blockchains.

#### *4.3.1. Network sharding*

The following schemes employ only network sharding in both permissionless and permissioned blockchain network.

***Domain and Static Sharding:*** Yoo et al. [81] proposed a permissioned blockchain framework for domain-based sharding, which partitions the network into multiple shards based on domains (regions), and each shard manages an individual domain. It can tolerate no more than 33% malicious nodes with the PBFT consensus. The overhead of intra-shard communication increases quadratically with the shard size. The ideology of geographical node allocation without the support of **RD** can be applied to other blockchain systems, thus fulfilling the criterion of **Un**. Unfortunately, there is no consideration on **IN**, **Co** and **PP**.

#### *4.3.2. Network sharding and transaction sharding*

The following schemes employ network sharding and transaction sharding in both permissionless and permissioned blockchain network.

***Fleetchain:*** Liu et al. proposed Fleetchain to reduce cross-shard communication overhead [82]. It constructs a fast byzantine fault tolerance (FBFT) intra-shard consensus mechanism and a responsive sharding transaction processing (RSTP) cross-shard consensus based on a robust (t, u)-multi-signature protocol and a Two Phase Commitment (2PC) protocol [109]. Furthermore, it can tolerate up to 33% byzantine nodes and processes UTXO-based transactions in a partial synchronous network. Its intra-shard communication overhead is proportional to the square of the shard size, while the complexity of cross-shard communication overhead is $\mathcal{O}(n^2+m)$. Fleetchain can be applied into both permissionless blockchain and permissioned blockchain, thus satisfying the criterion of **Un**. However, the fulfillment of **RD** cannot be justified as the node allocation process is not mentioned. Besides, it cannot support **Co**, **IN** and **PP**.

***Polyshard:*** Polyshard [83] is a polynomial coding sharding scheme that utilizes a Lagrangian-encoded computation framework to calculate and store encoded transactions to save storage space. To prevent erroneous results from malicious nodes, Polyshard employs noisy polynomial interpolation techniques, such as Reed–Solomon decoding [110]. This approach significantly improves storage efficiency, with the latency of Polyshard increasing linearly with running time, shard size, and shard number. Its throughput also grows linearly with the total number of nodes in the network. This scheme satisfies the criterion of **Un** as it supports various types of blockchain systems and consensus mechanisms. However, the detail of node allocation is not provided. Therefore, it does not achieve **RD**. Meanwhile, this scheme cannot support **Co**, **IN** and **PP**.

***DQNSB:*** Yun et al. [84] proposed a sharding scheme called DQNSB on the basis of deep Q-learning algorithm to dynamically perform the optimal sharding configuration [84]. The scheme applied PBFT consensus mechanism with 33% FTR to validate intra-shard transactions. Its throughput increases linearly when the number of nodes is less than 1000, but keeps steady with more nodes. Moreover, its latency always keeps constant and the intra-shard communication overhead increases quadratically with the shard size. By adopting the analysis equations of latency and a deep reinforcement learning (DRL) method [111], DQNSB dynamically finds the optimal throughput configuration for the massive IoT blockchain [112]. The application of DRL supports the criterion of **IN**. Regardless of the blockchain types, DQNSB can process account-based transactions within a variety of scenarios, thus reaching the criterion of **Un**. In addition, it performs adaptive node allocation with a PoW-ID mechanism, therefore it meets **RD**. Unfortunately, **PP** and **Co** were not explored.

#### *4.3.3. Network sharding, transaction sharding and state sharding*

The following schemes employ network sharding, transaction sharding and state sharding in both permissionless and permissioned blockchain network.

***OmniLedger:*** On the basis of ELASTICO [19], KokorisKogias et al. [20] proposed a scale-out distributed ledger named OmniLedger. Composed of an identity chain and multiple shard chains, OmniLedger maintains the global state and is more efficient and secure. OmniLedger is deployed in a partially synchronous network and applies ByzCoinX as well as Atomix intra-shard consensuses to validate UTXO-based transactions, which can tolerate up to 25% byzantine nodes. Its latency is proportional to the number of nodes. Meanwhile, its throughput and cross-shard communication overhead increase linearly with the number of shards. In addition, its intra-shard communication overhead grows logarithmically with the shard size. It provides a bias-resistant distributed random generation protocol by combining RandHound [98] with a VRF-based leader election algorithm [113], meeting the criterion of **RD**. It is compatible with various consensus mechanisms and can be applied to permissioned and permissionless blockchain systems, which implies that it satisfies **Un**. Moreover, ELASTICO cannot support **IN**, **PP** and **Co**.

***Ostraka:*** Manuskin et al. [85] proposed a scaling node architecture in a non-democratic environment, called Ostraka, in which a node can join in different shards simultaneously. Its latency and throughput increase linearly and logarithmically with the number of shards respectively. Meanwhile, it can be used in conjunction with diverse consensus mechanisms in any system with uneven distribution of voting power. Thus, it achieves **Un**. However, **RD** was not considered in Ostraka. In addition, **IN**, **PP** and **Co** were not explored.

***State Sharding with Space-aware Representations:*** Mizrahi and Rottenstreich proposed a traffic-aware sharding system based on a memory-light mapping algorithm to decrease cross-shard communication overhead by grouping together parts of the system state that are frequently validated [86]. It can be applied to both permissionless and permissioned blockchain systems without the requirement of using a specific consensus mechanism, thus satisfying **Un**. However, **RD** was not mentioned. Other criteria like **Co**, **IN** and **PP** were overlooked.

## 5. Open issues and future research directions

Based on the analysis and comparison of the reviewed schemes in Section 4, we list a number of open issues and further propose several future research directions.

### *5.1. Open issues*

In this part, we propose a number of open issues based on the above comprehensive review and analysis in terms of communication overhead, synchronization, automation, universality, and intelligence, as well as privacy preservation.

**High communication overhead:** Sharding still incurs communication overhead although the throughput of blockchains applying sharding has been significantly improved compared with traditional blockchain systems such as Bitcoin and Ethereum. However, sharding establishment results in extra computation and communication costs due to transaction broadcast and data distribution. On one hand, as the data in each shard is independent of each other, it is necessary to execute cross-shard communication to obtain data in other shards or achieve a consistent global state. However, cross-shard communication in turn has a severe effect on the throughput of a blockchain system. When the number of cross-shard transactions increases, the system throughput drops significantly. How to reduce communication overhead by avoiding cross-shard communications but still ensuring atomicity is still an open issue. On the other hand, intra-shard consensus mechanism costs significant resources such as computing power, network bandwidth, and storage space, which are scarce in a blockchain system. Especially, it may introduce extra communication overhead to reach cross-shard consensus.

**Ignorance of sharding synchronization:** Sharding synchronization is seldom considered in the current research, especially in the context of an asynchronous network. Although existing sharding schemes can improve the throughput of a blockchain network by processing transactions in parallel, few of them consider the fact that a blockchain system typically operates in an asynchronous network. Actually, it is intricate to design an effective sharding scheme in the asynchronous network, in which nodes have different views of the global state of the network. This greatly impacts efficiency consensus and makes atomicity hard to be achieved.

**Lack of automatic sharding:** The current research results cannot well support automatic sharding with both efficiency and security. Smart contracts are originally intended to eliminate the need of depending on a third party. Thereby, using the smart contract can automate the execution of an agreement without any intermediary's involvement or time loss. It enables a set of complex digital commitments to be correctly executed with conditions according to previous agreements. However, existing schemes only apply smart contracts to process transactions, rather than automatically partitioning nodes or transactions during the sharding establishment process with efficiency, stability and security.

**Poor universality:** The universality of existing sharding schemes is still inadequate. Sharding schemes are preferred to adapt to different application scenarios, which implies sound applicability and low maintenance cost. However, the universality of existing sharding schemes cannot meet the demand of sharding development. On one hand, most existing researches focus on a specific scenario, which limits their widespread application. On the other hand, even though some works such as Fleetchain [82], Repchain [66], Pyramid [70] and SharPer [79] attempt to build a universal sharding framework, their throughputs always suffer from a bottleneck as the number of nodes increases.

**Lack of Intelligence:** The current sharding establishment lacks intelligence in node and transaction allocation. Based on our review and analysis, it is obvious that node allocation and transaction assignment have a significant impact on the efficiency of sharding. Unfortunately, most studies apply fixed configurations to initialize sharding in different scenarios, in which sharding performance cannot be effectively guaranteed in an adaptive way. On the other hand, existing optimization schemes regarding node and transaction allocation are not perfect. They only consider local optimization rather than global optimization. Thus, making sharding intelligent enough to fit into different scenarios and ensure optimized performance remains an open and interesting issue [114].

**No privacy preservation:** Privacy preservation in sharding is rarely considered or successfully solved in current literature. The identities of users and details of transactions recorded on the blockchain are publicly available. Besides, the process of invoking a smart contract can expose the identity of a caller. However, few existing sharding schemes consider privacy preservation. In addition, existing solutions either request the support of TEE, or rely on time-consuming cryptographic schemes, which are not suitable for practical deployment. As a result, there is a need for serious study on privacy-preserving sharding in the literature.

**Security deficiency:** Each type of sharding technique has its own shortcomings. Network sharding introduces security concerns because nodes are partitioned randomly without considering their heterogeneity, such as different trustworthiness [115]. Moreover, the lack of consideration on the capability of each node has led to a significant disparity in average node computing power between each shard. This may result in an uneven workload distribution during transaction processing among shards, which impacts the overall performance of the blockchain system. Transaction sharding, on the other hand, introduces additional communication overhead due to cross-shard validation to deal with the double-spending problem. Besides, transactions may be processed in different shards and in different orders, which may cause inconsistency and conflicts in the blockchain system. What is more, it is important to ensure that the nodes do not collide with each other to manipulate the blockchain system. State sharding requires additional storage cost due to the backup to maintain the complete global state of the blockchain. Moreover, a centralized backup approach suffers from a single point of failure, which introduces additional security concerns as shards can be targeted by attackers to perform double-spend attacks or other types of attacks.

**Open issues of sharding schemes in permissioned and permissionless blockchain:** Sharding research faces distinct challenges depending on the type of blockchain. In permissioned blockchains, participants require authorization from a centralized manager to gain access to the network, which is also responsible for creating and managing shards. As a result, centralization-related security issues such as single points of failure could raise. In permissionless blockchains, the management of shards becomes complex due to complete decentralization. Additionally, data privacy introduces additional challenges in permissionless blockchains compared to permissioned ones. The primary concern in permissioned blockchains is ensuring data privacy while also achieving effective sharding communications. Current solutions for data privacy, such as encryption and access control, negatively impact performance and scalability. On the other hand, preventing attacks on permissionless blockchains raised by malicious nodes in different shards is a significant challenge. As the number of shards increases, the threats caused by attacks also amplify. Ensuring interoperability between different shards without sacrificing security or efficiency is a common issue of both types of blockchains. Existing solutions rely heavily on complex consensus protocols that may pose implementation challenges and lead to decreased performance.

### 5.2. Future directions

In this part, we suggest a series of potential future research directions based on the above open research issues.

**Efficient sharding with low communication overhead:** In the coming future, efforts should be focused on studying efficient sharding techniques with low communication overhead. In the process of consensus determination, cross-shard transactions with existing consensus mechanisms always suffer from high communication overhead, which has a severe impact on the throughput of a blockchain system. Therefore, a new consensus mechanism should be investigated to effectively decrease the cross-shard communication overhead while ensuring other quality properties. Additionally, a communication-efficient consensus mechanism should be explored to reduce the communication cost of block consensus. Meanwhile, advanced compression and encoding techniques can be investigated to minimize the amount of data that need to be transmitted within and among shards.

**Sharding automation with smart contract:** Smart contract-supported sharding systems offer automation and enhance security, but improving efficiency remains a challenging issue that requires further study. We should seriously explore smart contract-supported sharding systems since the introduction of smart contracts provides automation and eliminates the need for a trusted third party. Meanwhile, how to appropriately invoke smart contracts to avoid unnecessary communication overhead in cross-shard communication should be seriously studied.

**Universal sharding system investigation:** Researching universal sharding systems that can be applied to various scenarios is highly anticipated. A universal sharding system has high applicability and low maintenance cost, thus easy to be deployed in practice. However, how to ensure high throughput of a sharded blockchain system when achieving universality is a difficult research problem. A universal sharding system still requires high performance in order to gain acceptance.

**Efficient privacy preservation:** The privacy preservation in sharded blockchain systems is a crucial research topic, as few existing schemes effectively achieve this goal. However, designing a lightweight scheme that does not significantly impact system performance is a challenging task. Obviously, privacy preservation should not significantly impact the performance of a sharded blockchain system with regard to throughput and latency.

**Intelligent sharding with optimized trust and security:** Intelligent sharding with optimized trust and security is an interesting research topic worthy of special study [116,117]. Due to transaction volume and blockchain network topology changes, sharding should not be static. With intelligence, sharding can be adaptively adjusted to make the blockchain achieve the best and expected performance different application scenarios and contexts. Meanwhile, security and trust evaluation mechanisms should be embedded into sharding during its establishment and configuration in order to intelligently optimize shard trust and security [118,119]. How to automatically optimize blockchain performance in different scenarios through intelligent sharding while considering security and trust in an efficient and sensitive way is an interesting research issue that is worth our efforts [120].

**Continuous exploration on different types of sharding techniques:** Future research on sharding techniques should aim to address the shortcomings of existing techniques. For network sharding, researchers should explore more sophisticated algorithms to partition nodes based on their heterogeneity, such as their trustworthiness and capabilities. This helps in ensuring that each shard contains reliable and sufficiently powerful nodes, while also balancing the workload distribution as much as possible. Transaction sharding can benefit from the development of communication-efficient cross-shard protocols that can prevent double spending, which can reduce the overhead associated with transaction processing in different shards and minimize inconsistency and conflicts in the blockchain system [121]. In addition, further research is needed to enhance the security of sharding techniques and prevent collusion between nodes [122]. State sharding research should focus on reducing storage costs and developing new security mechanisms to protect against double-spending attacks or other targeted shard attacks. Investigating secure backup methods with low storage load or exploring lightweight backup methods is essential. Overall, future research should strive to make sharding techniques more efficient, secure, and scalable, so that they can be effectively applied to current and emerging blockchain systems.

**Future research directions of sharding schemes in permissioned and permissionless blockchains:** Future research should prioritize improving the efficiency and security of secure shard management in both permissioned and permissionless blockchains [123]. One potential avenue for exploration is the development of novel mechanisms that can ensure shard management and interoperability without compromising security or efficiency. Another potential direction is to investigate data privacy-enhancing techniques that do not negatively impact performance or scalability of permissioned blockchains. In addition, designing effective strategies to prevent attacks on blockchain raised by malicious nodes in shards, especially as the number of shards increase, is another pertinent research topic. Finally, exploring ways to mitigate centralization-related security issues and single points of failure in permissioned blockchains would be a significant effort in the field of new blockchain architecture.

## 6. Conclusion

This paper provided a thorough review on the state-of-the-art sharding techniques. We classified existing sharding schemes based on supported blockchain types and applied sharding techniques. In order to evaluate the performance of existing sharding schemes in a uniform way, we proposed a series of evaluation criteria in terms of scalability, applicability, and reliability. Furthermore, we conducted a comprehensive review on existing sharding schemes and seriously analyzed their advantages and disadvantages by adopting the proposed criteria. Through a serious and insightful review on cutting-edge sharding schemes, we explored open research issues and pointed out several potential future research directions to attract special efforts and investigation.

## 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.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62072351 and Grant 62202359; in part by the Key Research Project of Shaanxi Natural Science Foundation under Grant 2023-JC-ZD-35; in part by the open research project of ZheJiang Lab under grant 2021PD0AB01.

## References

[1] Knirsch F, Unterweger A, Engel D. Implementing a blockchain from scratch: why, how, and what we learned. EURASIP J Inf Secur 2019;2019:1–14.

[2] Liang X, Yan Z, Kantola R. GAIMMO: A grade-driven auction-based incentive mechanism with multiple objectives for decentralized crowdsourcing. IEEE Internet Things J 2022;9(18):17488–502.

[3] Wu Y, Yan Z, Yu FR, Deng R, Varadharajan V, Chen W. Guest editorial: Blockchain and healthcare computing. IEEE J Biomed Health Inform 2020;24(8):2144–5.

[4] Monrat AA, Schelén O, Andersson K. A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access 2019;7:117134–51.

[5] Han H, Fei S, Yan Z, Zhou X. A survey on blockchain-based integrity auditing for cloud data. Digit Commun Netw 2022;8(5):591–603.

[6] Sanka AI, Cheung RC. A systematic review of blockchain scalability: Issues, solutions, analysis and future research. J Netw Comput Appl 2021;195:103232.

[7] Wu Y, Yan Z, Thulasiram RK, Atiquzzaman M. Special issue on blockchain in future networks and vertical industries. IEEE Trans Netw Sci Eng 2021;8(1):1117–9.

[8] Han P, Yan Z, Ding W, Wan Z. A survey on cross-chain technologies, ACM distributed ledger technologies: Research and practice. 2022.
[9] Francisco R. Transactions per second (TPS). 2022, https://academy.binance.com/en/glossary/transactions-per-second-tps.
[10] Cao B, Zhang Z, Feng D, Zhang S, Zhang L, Peng M, et al. Performance analysis and comparison of pow, PoS and DAG based blockchains. Digit Commun Netw 2020;6:480–5.
[11] Gervais A, Karame GO, Capkun V, Capkun S. Is bitcoin a decentralized currency. IEEE secur priv 2014;12:54–60.
[12] Bagui S, Nguyen LT. Database sharding: to provide fault tolerance and scalability of big data on the cloud. Int J Cloud Appl Comput (IJCAC) 2015;5:36–52.
[13] Hafid A, Hafid AS, Samih M. New mathematical model to analyze security of sharding-based blockchain protocols. IEEE Access 2019;7:185447–57.
[14] Yu G, Wang X, Yu K, Ni W, Zhang JA, Liu RP. Survey: Sharding in blockchains. IEEE Access 2020;8:14155–81.
[15] Wang G, Shi ZJ, Nixon M, Han S. Sok: Sharding on blockchain. In: Proceedings of the 1st ACM conference on advances in financial technologies. 2019, p. 41–61.
[16] Han R, Yu J, Lin H, Chen S, Esteves-Veríssimo P. On the security and performance of blockchain sharding. 2021, Cryptology ePrint Archive.
[17] Liu Y, Liu J, Salles MAV, Zhang Z, Li T, Hu B, et al. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. Comput Sci Rev 2022;46:100513.
[18] Zhou Q, Huang H, Zheng Z, Bian J. Solutions to scalability of blockchain: A survey. IEEE Access 2020;8:16440–55.
[19] Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P. A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016, p. 17–30.
[20] Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In: 2018 IEEE symposium on security and privacy. 2018, p. 583–98.
[21] Zamani M, Movahedi M, Raykova M. Rapidchain: Scaling blockchain via full sharding. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018, p. 931–48.
[22] Wang J, Wang H. Monoxide: Scale out blockchains with asynchronous consensus zones. In: 16th USENIX symposium on networked systems design and implementation. 2019, p. 95–112.
[23] Xie J, Yu FR, Huang T, Xie R, Liu J, Liu Y. A survey on the scalability of blockchain systems. IEEE Netw 2019;33:166–73.
[24] Kim S, Kwon Y, Cho S. A survey of scalability solutions on blockchain. In: 2018 International conference on information and communication technology convergence. 2018, p. 1204–7.
[25] Chauhan A, Malviya OP, Verma M, Mor TS. Blockchain and scalability. In: 2018 IEEE international conference on software quality, reliability and security companion. 2018, p. 122–8.
[26] Scherer M. Performance and scalability of blockchain networks and smart contracts. 2017.
[27] Cope J. What's a peer-to-peer (P2P) network. 2002, https://www.computerworld.com/CONFERENCE/2588287/networking-peer-to-peer-network.html.
[28] Urquhart A. The inefficiency of bitcoin. Econ Lett 2016;148:80–2.
[29] Davis J. The crypto-currency: Bitcoin and its mysterious inventor. 10, 2011, The New Yorker.
[30] Xie H, Fei S, Yan Z, Xiao Y. SofitMix: A secure offchain-supported bitcoin-compatible mixing protocol. IEEE Trans Depend Secure Comput 2022;1–15.
[31] Biktimirov M, Domashev A, Cherkashin P, Shcherbakov AY. Blockchain technology: Universal structure and requirements. Automat Document Math Linguist 2017;51:235–8.
[32] Becker G. Merkle signature schemes, merkle trees and their cryptanalysis. Tech. rep, 12, Ruhr-University Bochum; 2008, p. 19.
[33] Blockchain architecture: State machine. 2021, https://docs.cosmos.network/v0.46/intro/sdk-app-architecture.html.
[34] Helliar CV, Crawford L, Rocca L, Teodori C, Veneziani M. Permissionless and permissioned blockchain diffusion. Int J Inf Manag 2020;54:102136.
[35] Bakos Y, Halaburda H, Mueller-Bloch C. When permissioned blockchains deliver more decentralization than permissionless. Commun ACM 2021;64:20–2.
[36] Marvin R. Blockchain: The invisible technology that's changing the world, PC MAG Australia. ZiffDavis, LLC. 25, 2017, Archived from the original on.
[37] Delgado-Segura S, Pérez-Sola C, Navarro-Arribas G, Herrera-Joancomartí J. Analysis of the bitcoin utxo set. In: International conference on financial cryptography and data security. 2018, p. 78–91.
[38] Chakravarty MM, Chapman J, MacKenzie K, Melkonian O, Peyton Jones M, Wadler P. The extended UTXO model. In: International conference on financial cryptography and data security. 2020, p. 525–39.
[39] Farrugia S, Ellul J, Azzopardi G. Detection of illicit accounts over the ethereum blockchain. Expert Syst Appl 2020;150:113318.
[40] Flora S. utxo vs account/balance model. 2018, https://medium.com/@sunflora98/utxo-vs-account-balance-model-5e6470f4e0cf.
[41] Shorish J. Blockchain state machine representation. 2018.
[42] Karame G. On the security and scalability of bitcoin's blockchain. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016, p. 1861–2.
[43] Nasir MH, Arshad J, Khan MM, Fatima M, Salah K, Jayaraman R. Scalable blockchains—A systematic review. Future Gener Comput Syst 2022;126:136–62.
[44] Wang X, Wang W, Zeng Y, Yang T, Zheng C. A state sharding model on the blockchain. Cluster Computing 2022;25:1969–79.
[45] Ittai A. Synchrony, asynchrony and partial synchrony. 2019, https://decentralizedthoughts.github.io/2019-06-01-2019-5-31-models/.
[46] Drăgoi C, Henzinger TA, Zufferey D. PSync: a partially synchronous language for fault-tolerant distributed algorithms. ACM SIGPLAN Notices 2016;51:400–15.
[47] Xu X, Sun G, Luo L, Cao H, Yu H, Vasilakos AV. Latency performance modeling and analysis for hyperledger fabric blockchain network. Inf Process Manag 2021;58:102436.
[48] Kuzlu M, Pipattanasomporn M, Gurses L, Rahman S. Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In: 2019 IEEE international conference on blockchain (Blockchain). 2019, p. 536–40.
[49] Costan V, Devadas S. Intel SGX explained, Cryptology ePrint Archive. 2016.
[50] Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: what it is, and what it is not. In: 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1. 2015, p. 57–64.
[51] Liu G, Yan Z, Feng W, Jing X, Chen Y, Atiquzzaman M. SeDID: An SGX-enabled decentralized intrusion detection framework for network trust evaluation. Inf Fusion 2021;70:100–14.
[52] Lin G, Wen S, Han Q-L, Zhang J, Xiang Y. Software vulnerability detection using deep neural networks: a survey. Proc. IEEE 2020;108(10):1825–48.
[53] Röscheisen M, Baldonado M, Chang K, Gravano L, Ketchpel S, Paepcke A. The stanford InfoBus and its service layers: Augmenting the internet with higher-level information management protocols. Digit Libr Comput Sci: MeDoc Approach 1998;213–30.
[54] Alharby M, Van Moorsel A. Blockchain-based smart contracts: A systematic mapping study. 2017, arXiv preprint arXiv:1710.06372.
[55] Wang M, Zhao D, Yan Z, Wang H, Li T. XAuth: Secure and privacy-preserving cross-domain handover authentication for 5G HetNets. IEEE Internet Things Journal 2022.
[56] Tanwar S, Bhatia Q, Patel P, Kumari A, Singh PK, Hong W-C. Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward. IEEE Access 2019;8:474–88.
[57] Chen X, Li C, Wang D, Wen S, Zhang J, Nepal S, et al. Android HIV: A study of repackaging malware for evading machine-learning detection. IEEE Trans Inf Forens Secur 2019;15(1):987–1001.
[58] Zhang J, Pan L, Han Q-L, Chen C, Wen S, Xiang Y. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. IEEE/CAA J Autom Sin 2021;9(3):377–91.
[59] Feng Q, He D, Zeadally S, Khan MK, Kumar N. A survey on privacy protection in blockchain system. J Netw Computr Appl 2019;126:45–58.
[60] Bernabe JB, Canovas JL, Hernandez-Ramos JL, Moreno RT, Skarmeta A. Privacy-preserving solutions for blockchain: Review and challenges. IEEE Access 2019;7:164908–40.
[61] Johnson B. Fault-tolerant microprocessor-based systems. IEEE Micro 1984;4:6–21.
[62] Antonucci F, Figorilli S, Costa C, Pallottino F, Raso L, Menesatti P. A review on blockchain applications in the agri-food sector. J Sci Food Agricul 2019;99:6129–38.
[63] Dong X, Prateek S, Christel Q, Jia Y, Max K. The zilliqa project: A secure, scalable blockchain platform. 2020.
[64] Lee DR, Jang Y, Kim H. Poster: A proof-of-stake (PoS) blockchain protocol using fair and dynamic sharding management. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. 2019, p. 2553–5.
[65] Nguyen LN, Nguyen TD, Dinh TN, Thai MT. Optchain: optimal transactions placement for scalable blockchain sharding. In: 2019 IEEE 39th international conference on distributed computing systems. 2019, p. 525–35.
[66] Huang C, Wang Z, Chen H, Hu Q, Zhang Q, Wang W, et al. Repchain: A reputation-based secure, fast, and high incentive blockchain system via sharding. IEEE Internet Things J 2020;8:4291–304.
[67] Tao Y, Li B, Jiang J, Ng HC, Wang C, Li B. On sharding open blockchains with smart contracts. In: 2020 IEEE 36th international conference on data engineering. 2020, p. 1357–68.
[68] Chen H, Wang Y. SSChain: A full sharding protocol for public blockchain without data migration overhead. Pervasive Mob Comput 2019;59:101055.
[69] Vitalik B, Gavin W, Charles H, Anthony DI, Joseph L. Ethereum. 2022, https://ethereum.org/en/.

[70] Hong Z, Guo S, Li P, Chen W. Pyramid: A layered sharding blockchain system. In: IEEE INFOCOM 2021-IEEE conference on computer communications. 2021, p. 1–10.
[71] Rana R, Kannan S, Tse D, Viswanath P. Free2Shard: Adversary-resistant distributed resource allocation for blockchains. Proc ACM Measur Anal Comput Syst 2022;6:1–38.
[72] Ren Z, Cong K, Aerts T, de Jonge B, Morais A, Erkin Z. A scale-out blockchain for value transfer with spontaneous sharding. In: 2018 Crypto valley conference on blockchain technology. 2018, p. 1–10.
[73] Danezis G, Meiklejohn S. Centrally banked cryptocurrencies. 2015, p. 934–50, arXiv preprint arXiv:1505.06895.
[74] Dang H, Dinh TTA, Loghin D, Chang E, Lin Q, Ooi BC. Towards scaling blockchain systems via sharding. In: Proceedings of the 2019 international conference on management of data. 2019, p. 123–40.
[75] Feng X, Ma J, Miao Y, Meng Q, Liu Q, Li H. Pruneable sharding-based blockchain protocol. Peer-to-Peer Netw Appl 2019;12:934–50.
[76] Greg M, Ed E, Kenny R, Evan J, Aleksandr B, Ian B. RChain Whitepaper 2021 (ver0.1). 2017, https://rchain.coop/whitepaper.html.
[77] Al-Bassam M, Sonnino A, Bano S, Hrycyszyn D, Danezis G. Chainspace: A sharded smart contracts platform. 2017, arXiv preprint arXiv:1708.03778.
[78] Androulaki E, Cachin C, Caro AD, Kokoris-Kogias E. Channels: Horizontal scaling and confidentiality on permissioned blockchains. In: European symposium on research in computer security. 2018, p. 111–31.
[79] Amiri M, Agrawal D, El Abbadi A. Sharper: Sharding permissioned blockchains over network clusters. In: Proceedings of the 2021 international conference on management of data. 2021, p. 76–88.
[80] Zheng P, Xu Q, Zheng Z, Zhou Z, Yan Y, Zhang H. Meepo: Sharded consortium blockchain. In: 2021 IEEE 37th international conference on data engineering. 2021, p. 1847–52.
[81] Yoo H, Yim J, Kim S. The blockchain for domain based static sharding. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering. TrustCom/BigDataSE, 2018, p. 1689–92.
[82] Liu Y, Liu J, Li D, Yu H, Wu Q. Fleetchain: A secure scalable and responsive blockchain achieving optimal sharding. In: International conference on algorithms and architectures for parallel processing. 2020, p. 409–25.
[83] Li S, Yu M, Yang C, Avestimehr AS, Kannan S, Viswanath P. Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously. IEEE Trans Inf Forens Secur 2020;16: 249–61.
[84] Yun J, Goh Y, Chung J. DQN-based optimization framework for secure sharded blockchain systems. IEEE Internet Things J 2020;8:708–22.
[85] Manuskin A, Mirkin M, Eyal I. Ostraka: Secure blockchain scaling by node sharding. In: 2020 IEEE European symposium on security and privacy workshops. 2020, p. 397–406.
[86] Mizrahi A, Rottenstreich O. State sharding with space-aware representations. In: 2020 IEEE international conference on blockchain and cryptocurrency. ICBC, 2020, p. 1–9.
[87] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus Rev 2008;21260.
[88] Lamport L, Shostak R, Pease M. The Byzantine generals problem. In: Concurrency: The works of leslie lamport. 2019, p. 203–26.
[89] Syta E, Tamas I, Visher D, Wolinsky DI, Jovanovic P, Gasser L, et al. Keeping authorities 'honest or bust' with decentralized witness cosigning. In: 2016 IEEE symposium on security and privacy. 2016, p. 526–45.
[90] Castro M, Liskov B. Practical Byzantine fault tolerance and proactive recovery. ACM Trans Comput Syst (TOCS) 2002;20:398–461.
[91] Castro M, Liskov B, et al. Practical byzantine fault tolerance. In: OsDI. 99, 1999, p. 173–86.
[92] De Angelis S, Aniello L, Baldoni R, Lombardi F, Margheri A, Sassone V. PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. 2018.
[93] Shrier I, Platt RW. Reducing bias through directed acyclic graphs. BMC Med Res Methodol 2008;8:1–15.
[94] Han R, Yan Z, Liang X, Yang LT. How can incentive mechanisms and blockchain benefit with each other? a survey. ACM Comput Surv (CSUR) 2022;55(7):1–38.
[95] Kogias EK, Jovanovic P, Gailly N, Khoffi I, Gasser L, Ford B. Enhancing bitcoin security and performance with strong consistency via collective signing. In: 25th usenix security symposium. 2016, p. 279–96.
[96] Huang D, Ma X, Zhang S. Performance analysis of the raft consensus algorithm for private blockchains. IEEE Trans Syst Man Cybern 2019;50:172–81.
[97] Howard H. ARC: analysis of raft consensus. Tech. rep, University of Cambridge, Computer Laboratory; 2014.
[98] Syta E, Jovanovic P, Kogias EK, Gailly N, Gasser L, Khoffi I, et al. Scalable bias-resistant distributed randomness. In: IEEE symposium on security and privacy. 2017, p. 444–60.
[99] Sen S, Freedman MJ. Commensal cuckoo: Secure group partitioning for large-scale services. ACM SIGOPS Oper Syst Rev 2012;46:33–9.
[100] Smith C. Proof-of-stake (POS). 2022, https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.
[101] L'Ecuyer P. Random number generation. In: Handbook of computational statistics. 2012, p. 35–71.
[102] What is randomness. 2022, https://eth2.incessant.ink/book/06__building-blocks/02__randomness.html.
[103] Johnson S, Scarlata V, Rozas C, Brickell E, Mckeen F. Intel software guard extensions: EPID provisioning and attestation services, vol. 1. White paper, 2016, p. 119.
[104] Thomasian A, Ryu IK. Performance analysis of two-phase locking. IEEE trans softw Eng 1991;17:386.
[105] Chepurnoy A, Larangeira M, Ojiganov A. Rollerchain, a blockchain with safely pruneable full blocks. 2016, arXiv preprint arXiv:1603.07926.
[106] Bhaskar ND, Chuen DLK. Bitcoin mining technology. In: Handbook of digital currency. 2015, p. 45–65.
[107] How the consensus protocol impacts blockchain throughput. 2022, https://www.nec.com/en/global/insights/CONFERENCE/2020022520/index.html.
[108] Lamport L. Paxos made simple, ACM SIGACT news (distributed computing column) 32, 4 (whole number 121, December 2001). 2001.
[109] Samaras G, Britton K, Citron A, Mohan C. Two-phase commit optimizations in a commercial distributed environment. Distrib Parallel Databases 1995;3:325–60.
[110] Kopparty S, Ron-Zewi N, Saraf S, Wootters M. Improved decoding of folded reed-solomon and multiplicity codes. In: 2018 IEEE 59th annual symposium on foundations of computer science. 2018, p. 212–23.
[111] Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep reinforcement learning: A brief survey. IEEE Signal Process Mag 2017;34:26–38.
[112] Wu Y, Zhang N, Yan Z, Atiquzzaman M, Xiang Y. Guest editorial special issue on AI and blockchain powered IoT sustainable computing. IEEE Internet Things 2023;10(8):6531–4.
[113] Gilad Y, Hemo R, Micali S, Vlachos G, Zeldovich N. Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th symposium on operating systems principles. 2017, p. 51–68.
[114] Liu K, Yan Z, Liang X, Kantola R, Hu C. A survey on blockchain-enabled federated learning and its prospects with digital twin. Digit Commun Netw 2022.
[115] Liu G, Yan Z, Wang D, Wang H, Li T. DePTVM: Decentralized pseudonym and trust value management for integrated networks. IEEE Trans Depend Secure Comput 2023.
[116] Feng W, Yan Z, Yang LT, Zheng Q. Anonymous authentication on trust in blockchain-based mobile crowdsourcing. IEEE Internet Things J 2020;9(16):14185–202.
[117] Liu Y, Wang J, Yan Z, Wan Z, Jäntti R. A survey on blockchain-based trust management for internet of things. IEEE Internet Things J 2023;10(7):5898–922.
[118] Yan Z, Peng L, Feng W, Yang LT. Social-chain: Decentralized trust evaluation based on blockchain in pervasive social networking. ACM Trans Internet Technol (TOIT) 2021;21(1):1–28.
[119] Liu G, Dong H, Yan Z, Zhou X, Shimizu S. B4SDC: A blockchain system for security data collection in MANETs. IEEE Trans Big Data 2020;8(3):739–52.
[120] Feng W, Yan Z. MCS-chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain. Future Gener Comput Syst 2019;95:649–66.
[121] Feng W, Li Y, Yang X, Yan Z, Chen L. Blockchain based data transmission control for tactical data link. Digit Commun Netw 2021;7(3):285–94.
[122] Choo KKR, Yan Z, Meng W. Blockchain in industrial IoT applications security and privacy advances, challenges and opportunities. IEEE Trans Ind Inf 2020;16(6):4119–21.
[123] Peng L, Feng W, Yan Z, Li Y, Zhou X, Shimizu S. Privacy preservation in permissionless blockchain: A survey. Digital Commun Netw 2021;7(3):295–307.