A Survey of Consortium Blockchain and Its Applications

Xiaotong Chen 1,2, Songlin He 1,2, Linfu Sun 1,2, Yangxin Zheng 1,2 and Chase Q. Wu 3,*

1 School of Computing and Artificial Intelligence, Southwest Jiaotong University (SWJTU), Chengdu 610031, China; c0672@my.swjtu.edu.cn (X.C.); sohe@swjtu.edu.cn (S.H.); sunlf@vip.163.com (L.S.); zhengyx555@163.com (Y.Z.)
2 Sichuan Provincial Key Laboratory of Manufacturing Industry Chain Collaboration and Information Support Technology, Southwest Jiaotong University, Chengdu 610031, China
3 Department of Computer Science, New Jersey Institute of Technology (NJIT), Newark, NJ 07102, USA
* Correspondence: chase.wu@njit.edu

Abstract: Blockchain is a revolutionary technology that has reshaped the trust model among mutually distrustful peers in a distributed network. While blockchain is well-known for its initial usage in a public manner, such as the cryptocurrency of Bitcoin, consortium blockchain, which requires authentication of all involved participants, has also been widely adopted in various domains. Nevertheless, there is a lack of comprehensive study of consortium blockchain in terms of its architecture design, consensus mechanisms, comparative performance, etc. In this study, we aim to fill this gap by surveying the most popular consortium blockchain platforms and assessing their core designs in a layered fashion. Particularly, Byzantine fault tolerant (BFT) state machine replication (SMR) is introduced to act as a basic computational model of consortium blockchain. Then the consortium blockchain is split into the hardware layer, layer-0 (network layer), layer-I (data layer, consensus layer and contract layer), layer-II protocols, and application layer. Each layer is presented with closely related discussion and analysis. Furthermore, with the extraction of the core functionalities, i.e., robust storage and guaranteed execution, that a consortium blockchain can provide, several typical consortium blockchain-empowered decentralized application scenarios are introduced. With these thorough studies and analyses, this work aims to systematize the knowledge dispersed in the consortium blockchain, highlight the unsolved challenges, and also indicate the propitious avenues of future work.

Keywords: consortium blockchain; consensus mechanisms; P2P communications; transactions; storage; decentralized applications

1. Introduction

Blockchain has emerged and revolutionized the conventional trust model amongst mutually untrusted users in a decentralized network. As an infrastructure technology, the applications of blockchain are far beyond its initial usage in cryptocurrency [1]. Complementary to the message transmission network built by the Internet, blockchain plays the critical role of value transmission network atop the widely used Internet. Until now, the advancement of blockchain has mainly experienced three stages [2], i.e., stage 1.0 initialized with cryptocurrency is represented by Bitcoin [3,4], stage 2.0 in the financial domain is represented by Ethereum [5], which supports Turing-complete programming languages to execute pre-agreed logic via smart contract, the ongoing stage 3.0 refers to various application fields including decentralized finance (Defi) [6], Internet of Things (IoT) [7], cyber security [8], content delivery networks [9], healthcare [10], smart city [11], metaverse [12], etc., which meet diversified and more complex real-world demands. With the blueprint and the gradual stepping into web3.0 [13] where data with semantic meanings are interconnected in a decentralized manner, blockchain sustainably shows its great potential in reshaping trust amongst individuals, and therefore, it is worth retrospectively investigating its core functionalities and primary applications.
Blockchain can be defined as an immutable ledger to record transactions and is maintained by distrustful peer nodes in a distributed network [14]. The wide adoption of blockchain is essentially owed to its multiple advantageous security properties [14,15]. Specifically, the availability property ensures that the blockchain network stays available even though partial nodes become unreachable; the immutability property guarantees that the recorded transactions cannot be reverted assuming the number of simultaneously corrupted nodes is upper-bounded. This consistency property assures that all peer nodes in the blockchain network remain a globally consistent ledger upon invocation, and the accountability property enables to take some corresponding actions, e.g., monetary punishment, if any peer performs malicious activities; the provenance property indicates that blockchain provides tamper-proof information about the origin of data records. Several studies [16–18] extract the core functions of a blockchain and formalize the blockchain model of cryptography. By constructing the ideal functionality, the cryptographic blockchain model is built in the Generalized Universal Composability (UC) framework [19], yielding the essential result of “fair Multi-Party Computation (MPC) with public deposits” aiming to comprehensively specify and reason about the security of blockchain-empowered protocols, and facilitate the designing of decentralized applications atop blockchains. As illustrated in Figure 1, $F_{\text{blockchain}}[\text{succ}]$ defines a general-purpose append-only ledger implemented by common blockchain protocols [20].

<table>
<thead>
<tr>
<th>$F_{\text{blockchain}}[\text{succ}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Parameter: successor relationship $\text{succ} : {0,1}^* \times {0,1}^* \rightarrow {0,1}$</td>
</tr>
<tr>
<td>2. Upon receiving (“init”): Storage := $\emptyset$</td>
</tr>
<tr>
<td>3. Upon receiving (“read”, id): output $\text{Storage}[\text{id}]$, or $\perp$ if not found</td>
</tr>
</tbody>
</table>
| 4. Upon receiving (“write”, id, inp) from $P$:
  | 5. let val := $\text{Storage}[\text{id}]$, set to $\perp$ if not found
  | 6. if $\text{succ(val,inp)} = 1$ then $\text{Storage}[\text{id}] := \text{val} || (\text{inp}, P)$; output (“receipt”, id)
  | 7. else output (“reject”, id) |
| 8. Upon receiving (“€”, id, val):
  | 9. if val $\in \text{Storage}[\text{id}]$ then output true else output false |

**Figure 1.** Blockchain ideal functionality [20]. The parameter $\text{succ}$ models the validity check of a transaction, $P$ refers to a client, and inp means the input data submitted by $P$.

One primary thinking in designing a blockchain-based decentralized application lies in choosing a proper type of blockchain [8] in light of the concrete application setting. Specifically, there are usually three categories: the public/permissionless blockchain allows anyone to join in or leave at their discretion; the permissioned consortium blockchain requires the participants to be authorized before accessing the blockchain network and the servers constructing the blockchain network are provided by multiple organizations; the permissioned private blockchain differentiates from the consortium blockchain in that the peers constructing the blockchain network belong to one organization. In practice, consortium blockchains enable participants of common interests to authenticate and collaborate, thereby reducing the trust risk to some degree since the peers are known to each other. Moreover, consortium blockchains provide greater control over the governance and decision-making processes than public blockchains. Hence, this study examines the consortium blockchain and demonstrates the potential for enhancing various existing applications.

To comprehensively survey the consortium blockchain and its applications, we first provide a layered architecture, as depicted in Figure 2, and elaborate on the core designs and comparisons in each layer. In particular, the bottom layer is hardware which involves traditional physical servers, switches, routers and trusted hardware. The network layer (referred to as layer 0) considers the distributed setting where communication models,
services, and typical algorithms are investigated. Layer I relates to the blockchain itself, containing a data layer, a consensus layer and a smart contract layer. The layer II protocols mainly function to improve the scalability and performance of layer I. Based on these layers, blockchain empowers a wide range of applications. Such a layered architecture of blockchain works as a main guide to survey related techniques in later sections.

![Figure 2. Consortium blockchain relevant layers.](image)

**Contributions.** Overall, our contributions can be summarized as follows.

- We survey the consortium blockchain and present its core in a layered manner, thus aiming to comprehensively cover pertinent studies.
- We summarize the academic advancements and the usage in practical applications of consortium blockchain and suggest a few potential future research directions in this field for enhancing its design and practicalization.

The rest of the paper is structured as follows: Section 2 concretely presents the consortium blockchain regarding its computational model, typical platforms and relevant layers; Section 3 introduces the closely related decentralized applications atop consortium blockchain; Section 4 provides reflections about consortium blockchain and its potential research directions and we conclude in Section 5.

2. Consortium Blockchain

In this section, we introduce the computational model of a consortium blockchain and popular consortium blockchain platforms and then present the consortium blockchain in a layered fashion including the hardware layer, network layer, layer I and layer II protocols.

2.1. State Machine Replication

As a fundamental distributed computing model, state machine replication (SMR) [21] aims to provide an abstract state machine distributed over the network and replicated by many peers (or called replicas) [22]. In particular, as illustrated in Figure 3, an SMR protocol starts with an initial state $S_0$, every client (or called process) can submit a request containing several consecutive execution commands, e.g., $C = \{c_1, c_2, c_3\}$ and perceive a sequence of commits. Each commit refers to the execution state that is produced via the execution of the commands atop its previous commit. Obviously, as the execution process is deterministic, i.e., with the same input, an algorithm would always generate the same output, and every
peer node would reach the new identical state. From a global viewpoint, all the replicas transfer their state \( S^0 \) to \( S^1 \). It is worth pointing out that invalid commands would be rejected during execution. Typically a SMR protocol meets the following requirements [23]: (i) any honest replica \( r_i \) starts with the state \( S^0 \); (ii) The honest replicas \( r_1, r_2, \ldots \) applying the same sequence of execution operations \( \{ c_1, c_2, \ldots \} \) would all reach the new state \( S^j \), \( j \in \{1, 2, \ldots \} \). Such a process satisfies two key security properties: (i) \textit{liveness}, every submitted valid command would be added to the commit; (ii) \textit{consistency}, all peer nodes perceive the same commits even though there is a communication delay.

![Figure 3. Illustration of state machine replication model.](image)

Blockchain is essentially a state machine where transactions are submitted by peer nodes in the blockchain network and the distributed ledger records the transactions and the latest state upon transaction execution in a consistent and finalized manner. In a consortium blockchain, the distributed SMR ensures consensus, fault tolerance and fairness. For instance, guaranteeing consensus can be realized by propagating the status of a contract to all nodes in the network. Furthermore, SMR in the blockchain setting is highly pertinent to Byzantine fault tolerance (BFT), i.e., the peers can reach consensus even in the presence of adversaries who may corrupt or control part of the replicas. Typical BFT SMR protocols include Practical Byzantine Fault Tolerance (PBFT) [24] and its descendants [25–32]. However, there are several subtle but important differences [23] between the BFT SMR approach and (consortium) blockchain: (i) blockchain applications require maintaining a verifiable persistent ledger containing the executed transactions and also supporting reconfigurations on the replicas; these two features are not present in the SMR implementations; (ii) though most literature about BFT SMR assumes a static set of processes, in a blockchain consortium, peer nodes are expected to be allowed to join and leave at any time, without the need for an additional trusted party.

2.2. Consortium Blockchain Platforms

Choosing a proper consortium blockchain platform can greatly improve the efficiency and practicality of a blockchain-enabled application. To this end, we first analyze the core functionalities of a consortium blockchain and present the designs of each layer with detailed analysis. We select the representative consortium blockchain platforms based on several relevant sources [33–37], which indicate the popularity and practicality of these platforms in various domains.

2.2.1. Core Functionalities of Consortium Blockchain

**Consortium Blockchain-Enabled Robust Storage.** One core functionality empowered by consortium blockchain lies in providing a robust storage capability. From such a perspective, it resembles a traditional distributed database which has been well-studied for decades. However, there are several key [8,38] differences between a consortium blockchain and a database system:

- **Trust model.** Traditional database systems require trusting all participants where even the malicious behaviors occurring from only one node can make the whole
system collapse; a blockchain system can tolerate partial, e.g., one-third of the nodes misbehaving arbitrarily. Essentially, a consortium blockchain-enabled storage system supports more robust byzantine fault tolerance instead of merely crash fault tolerance.

- **Transaction processing.** Though most consortium blockchain systems also support parallel transaction processing capability, they differ from the database systems in several aspects: (i) consortium blockchain commits transactions at the block level while the database commits at the individual transaction level; (ii) consortium blockchain is a distributed system where the state is fully replicated across the network and the transaction operates on all nodes, while for a database system, the transactions usually operate on a subset of network nodes; (iii) during transaction execution, the state on different nodes may differ in a consortium blockchain network, while for a database, a transaction is executed once against the only state present in the system; (iv) the factors dominating the performance of a consortium blockchain and a database is distinct, i.e., cryptographic primitive computations, network communication for the former, and the locking mechanism of concurrency control for the latter.

It is strongly believed that transitioning the technology of storage from a traditional database system to a consortium blockchain is the right direction, and we have seen several works [38,39] that are dedicated to research in such a field. Consortium blockchains indeed possess the potential to improve storage robustness. However, the storage cost is huge as all nodes need to store a full copy of the data submitted to the consortium blockchain network. A group of work also concentrated on reducing the on-chain storage costs, as tabulated in Table 1.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Literature</th>
<th>Highlights</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharding</td>
<td>Wang et al. [40]</td>
<td>Sharding Technology</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Wu et al. [41]</td>
<td>KBFT</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Shen et al. [42]</td>
<td>A Node Reliable Shard Model</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Zheng et al. [43]</td>
<td>Replay-epoch &amp; Cross-call</td>
<td>2022</td>
</tr>
<tr>
<td></td>
<td>Zhou et al. [44]</td>
<td>Dynamic Sharding</td>
<td>2020</td>
</tr>
<tr>
<td></td>
<td>Qi et al. [45]</td>
<td>Erasure Coding</td>
<td>2020</td>
</tr>
<tr>
<td>Compression</td>
<td>Farahat et al. [46]</td>
<td>The LZ4 Algorithm</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Karthik et al. [47]</td>
<td>Lempel-Ziv-Welch</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Yu et al. [48]</td>
<td>PoW-BC</td>
<td>2021</td>
</tr>
<tr>
<td>Deduplication</td>
<td>Liu et al. [49]</td>
<td>A New Refreshable Encryption Algorithm</td>
<td>2022</td>
</tr>
<tr>
<td>Compressed Indexing</td>
<td>Zhou et al. [50]</td>
<td>Merkle Semantic Trie</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Chen. et al. [51]</td>
<td>Index Pointers</td>
<td>2023</td>
</tr>
<tr>
<td></td>
<td>Shafarenko [52]</td>
<td>Tunstall’s Method</td>
<td>2022</td>
</tr>
</tbody>
</table>

**Consortium Blockchain-Enabled Guaranteed Computing.** Another core functionality empowered by consortium blockchain is the capability of guaranteed execution. Underpinned by the smart contract which can execute pre-determined programs without any interruption, consortium blockchain can faithfully execute any deployed executable code, e.g., business agreement [53], and thus provide the great potential to revolutionize many existing centralized applications [16].

2.2.2. Hyperledger Fabric

Hyperledger Fabric [54] is a popular open-source consortium blockchain system introduced by IBM. The typical message flow in Hyperledger Fabric is depicted in Figure 4, where a client submits transactions to a group of peer nodes called **endorsing nodes** or **endorsers**, which simulate the transaction execution and send back the endorsed results, then the signed/endorsed results are forwarded to the **ordering nodes** or **orderers**, which accumulate transactions into blocks and output blocks abiding by certain rules, e.g., following a certain time interval or an approximate block size. The generated blocks are sent
to committing nodes or committers, which would write to the distributed ledger upon the validation of the generated blocks pass. Note, that the orderers can either be the full nodes in the blockchain network or extra servers can be employed. Surrounding Hyperledger Fabric, research mainly focusing on the following aspects is conducted: (i) performance modeling and analysis [55–57]; (ii) privacy and security [58–60]; (iii) interoperability [34,61]; (iv) benchmark and visualization [57,62,63].

Figure 4. A typical transaction workflow in Hyperledger Fabric.

2.2.3. Ethereum

Ethereum [5] is blockchain 2.0 since it supports the Turing-complete smart contract, which enables the execution of much more complex logic in various application scenarios beyond cryptocurrency. In contrast to the unspent transaction output (UTXO) model in Bitcoin, Ethereum is designed with the account-based model where the latest balance for a user equipping with a public-private key pair is bound with his/her address (hash of the user’s public key). In addition, Ethereum has transitioned its consensus mechanism from Proof of Work (PoW) to Proof of Stake (PoS) in 2022, i.e., the so-called Merge [64], to significantly reduce the criticized energy cost. Ethereum is usually characterized by three features: (i) smart contract [65]; (ii) Ethereum virtual machine (EVM) [66] and (iii) decentralized applications (DAPPs) [67]. Apart from the wide application in public settings, Ethereum can also be deployed in a consortium [68] or private [69] manner. It is worth pointing out that Ethereum is popularized as a public blockchain protocol, yet it also supports the customization of building a consortium blockchain environment for diversified application scenarios [70,71].

2.2.4. FISCO BCOS

FISCO BCOS [36] is an enterprise-grade permissioned blockchain system developed by the Financial Blockchain Shenzhen Consortium which is compatible with Ethereum in terms of account management and smart contracts [72]. FISCO BCOS aims to significantly improve the consortium blockchain’s performance. Such an ambition is mainly achieved via two designs: a block level pipelining workflow designed to break the serial dependency of blocks; the blocks are processed in a pipeline with four stages. Blocks can be processed at different stages simultaneously. Meanwhile, a deterministic multi-contract mechanism is designed to execute transactions in parallel within a block. Transactions are also dispatched into multiple shards and processed in parallel by a set of executors. Consequently, FISCO BCOS realizes both intra-block and inter-block parallelism, and therefore, enables considerable performance and scalability enhancement [33].

2.2.5. Corda

Corda [73] is a distributed ledger platform designed specifically for the financial sector. It is utilized by over 60 companies, banks such as HSBC, J.P. Morgan, and institutions like
Intel, Microsoft and NASDAQ [74]. There exist several key differences [75] between Corda and conventional blockchains: (i) there are no transactions structured in blocks; (ii) the occurred transactions are only shared with involved parties instead of all participants; (iii) a set of trusted parties called notaries are introduced in order to prevent a double-spending attack. In light of these characteristics, Corda highlights three key properties:

- **Privacy.** Corda prioritizes privacy by design, ensuring that transactions are only shared on a need-to-know basis. Unlike many other blockchains, Corda achieves a weaker security notion *partial consistency* where parties in the blockchain network may only see part of the state but accumulating all parties’ states can result in the global view. This minimizes the exposure of sensitive data and reduces the network load and storage requirements [35].

- **Scalability.** Unlike many other blockchains, Corda does not rely on a global consensus mechanism that requires every node to validate every transaction but instead utilizes a pluggable notary service that can employ various consensus algorithms depending on the use case. Such a design allows Corda to handle high transaction volumes and complex business logic without compromising performance or security [76].

- **Interoperability.** Corda allows businesses to use their legacy infrastructure while benefiting from the advantages of distributed ledger technology. It also supports interoperability among different Corda networks, as well as with other blockchain platforms via the use of common standards and protocols. Such a design enables cross-industry and cross-border collaboration and innovation [77].

2.2.6. Quorum

Quorum [78] is an Ethereum-based, enterprise-focused, permissioned blockchain infrastructure specifically designed for financial use cases [37]. This open-source project was initiated by J.P. Morgan Chase and has been acquired by ConsenSys. Quorum contains two blockchain projects: one is based on GoQuorum [79], and the other one is based on Hyperledger Besu [80]. Each Quorum node is composed of two main services: (i) *Quorum client*, which is responsible for executing the Ethereum p2p protocol and the consensus algorithm; (ii) *privacy manager*, which enables private transactions and smart contract operations. As a consortium blockchain, Quorum is mainly introduced to satisfy the following demands [81]: (i) empowered by the privacy manager to execute private transactions and smart contract operations; (ii) supporting multiple pluggable consensus mechanisms; (iii) enabling flexible and expressive network permissions management.

2.2.7. Ripple

Ripple [82] is a real-time gross settlement (RTGS) system aiming at fast global payments, asset exchange, and settlement [83]. Ripple maintains a ledger of transactions where participants can trade user-issued currencies along with the native cryptocurrency of Ripple, i.e., XRP. The *Ripple protocol consensus algorithm* (RPCA) allows for a flexible security assumption for consensus protocols (typically tolerating $<1/3$ faulty nodes) in the sense that each node can declare which nodes it trusts instead of using a global assumption about how many faulty nodes may collude simultaneously and misbehave. In practical usage, a company that wants to employ the Ripple network can develop software and interact with it, e.g., SAP developed a Ripple-enabled application for cross-border payments between two banks which significantly decreased time costs, i.e., from six business days to only 20 s [84].

Table 2 presents the comparison of the aforementioned consortium blockchain platforms in terms of their data models, supported consensus mechanisms, state databases, highlighted properties and performance under specific experimental environments.
Table 2. The comparison of different consortium blockchain platforms.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Data Model</th>
<th>Consensus</th>
<th>State Database</th>
<th>Highlights</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperledger Fabric [54]</td>
<td>Account Based</td>
<td>Raft, PBFT</td>
<td>CouchDB, LevelDB</td>
<td>Pluggable Consensus, Scalability</td>
<td>4 nodes</td>
</tr>
<tr>
<td>Ethereum [85]</td>
<td>Account Based</td>
<td>PoW, PoS</td>
<td>MPT</td>
<td>Turing-Complete Smart Contract</td>
<td>10 nodes</td>
</tr>
<tr>
<td>FISCO BCOS [33]</td>
<td>Account Based</td>
<td>Raft, PBFT</td>
<td>LevelDB</td>
<td>Efficiency, Flexibility</td>
<td>6 nodes</td>
</tr>
<tr>
<td>Corda [86]</td>
<td>UTXO Based</td>
<td>Raft, PBFT</td>
<td>H2</td>
<td>Privacy, Scalability</td>
<td>4 node</td>
</tr>
<tr>
<td>Quorum [87]</td>
<td>Account Based</td>
<td>Raft, PBFT</td>
<td>Go-Ethereum</td>
<td>Modularity, Privacy</td>
<td>3 nodes</td>
</tr>
<tr>
<td>Ripple [88]</td>
<td>Ripple Data Model</td>
<td>XRP</td>
<td>Rippled Database</td>
<td>RTGS, Native Token</td>
<td>16 nodes</td>
</tr>
</tbody>
</table>

2.3. Hardware Layer

The hardware layer refers to the underlying hardware equipment and infrastructure, including computers, servers, network equipment and so forth. Especially, trusted hardware provides a trusted execution environment (TEE) [89], i.e., a hardware architecture that enables code execution in an isolated, tamper-free environment (called a secure enclave) [90]. A secret key hidden in the enclave and possibly only known to the hardware manufacturer can be used to encrypt incoming and outgoing data, thus ensuring data confidentiality. Meanwhile, TEE can attest that an output represents the result of code execution, and allows remote users to make sure the execution is correct. In recent years, TEEs designed with a secure counter and supporting complex stateful functions are more preferred, popular products including Intel SGX [91], Intel TXT [92], ARM’s TrustZone [93], AMD SEV [94], Sanctum [95], KeyStone [96], etc. TEEs exhibit a set of security features [97,98], and therefore, are adopted in many complicated system designs [99]. However, TEEs are also vulnerable to a few attacks, e.g., side-channel attacks [100] and rewind attacks [101], and plenty of work [98,102–104] focuses on such a direction to defend against potential attacks. The hardware layer provides the physical resources and support required for the operation of the consortium blockchain systems and plays a crucial role in affecting the performance, security and reliability.

2.4. Network Layer

The network layer not only includes the complete network stack of the conventional network architecture which concentrates on Internet routing, but it also forms a dedicated peer-to-peer network for blockchain nodes. This layer is of great importance since it impacts the scalability [105], security [106] and privacy [107] of a blockchain network. In a consortium blockchain network, full nodes are required to participate in consensus, and also periodically communicate with each other to maintain the connection [108], e.g., via gossip protocol [7]. When designing consortium blockchain-enabled decentralized applications, it is necessary to consider different communication models, e.g., synchronous, partial synchronous and asynchronous. A main assumption regarding the network layer is that such a layer should provide reliable communication among peer nodes in a blockchain network.

2.5. Layer I: Data, Consensus Mechanism, and Smart Contract

Layer I refers to consortium blockchain per se (As illustrated in Figure 2, layer-I refers to the design of the consortium blockchain itself regardless of the underlying hardware, network environment, or other protocols that are designed atop consortium blockchain). It hosts an append-only chain of blocks that accumulate transactions in the blockchain network for public verifiability [109]. It can further be categorized into the sub-layers, i.e., the datalayer, consensus layer and contract layer.
2.5.1. Data Layer

The data layer mainly concerns the data structure and data storage. Specifically, blockchain is typically a succession of blocks with a starting block called *genesis block*. Transactions in each block can simply be a transaction list, e.g., in Bitcoin, or a more intricate structure such as state trie in Ethereum. For the consortium blockchain, we highlight the following components in the data layer.

- **Block.** A block contains two parts, i.e., the *block head* and the *block body*, where the head part typically includes the block version, the merkle root of the involved transactions, timestamp, nonce and the hash of the previous block. The body part is mainly composed of a transaction counter and a bunch of transactions. The number of transactions is related to the block size, which is restricted due to the communication overhead. Meanwhile, asymmetric cryptography, i.e., digital signatures such as the *elliptic curve digital signature algorithm* (ECDSA) [110] is used to ensure the validity of transactions, where usually the digital signature requires *existential unforgeability under chosen message attack* (EU-CMA) security [111].

- **Chain Structure.** From the data structure viewpoint, the architecture of a consortium blockchain is essentially a hash chain where the unique hash value of each block is computed based on its previous one. Such a design fully hinges on the security properties such as one-way, collision resistance of the hash function \( H : \{0,1\}^* \rightarrow \{0,1\}^\lambda \) with the security parameter \( \lambda \), and can further be modeled as a random oracle [112]. In addition, to improve the scalability and reduce the latency of the hash-chain based blockchain system, some works [113–115] have explored the *directed acyclic graphs* (DAG) architecture of blockchain, exemplified by IOTA [116], an open source distributed ledger designed for the IoT.

- **Merkle Tree.** A Merkle tree MT is constructed from the leaf nodes level all the way up to the Merkle root level by grouping nodes in pairs and calculating the hash of each pair of nodes in that particular level [117]. Specifically, the MT scheme contains a tuple of three algorithms (BuildMT, GenMTP, VerifyMTP), as illustrated in Algorithms 1–3. BuildMT accepts as input a sequence of elements \( m := (m_1, m_2, \ldots, m_n) \), and outputs the Merkle tree MT with root. GenMTP takes as input the Merkle tree MT and the hash of the \( i \)-th message in \( m \), i.e., \( H(m_i) \), and outputs a proof \( \pi \) to attest the inclusion of \( m_i \) at the position \( i \) of \( m \). VerifyMTP takes as input the Merkle tree proof \( \pi \), the root of MT and the message hash \( H(m_i) \), and outputs either *true* or *false* indicating whether it succeeds in verifying or not. The security of the Merkle tree scheme ensures that: for any *probabilistic polynomial-time* (P.P.T.) adversary \( A \), any sequence \( m \) and any index \( i \), conditioned on MT is a Merkle tree built for \( m \), \( A \) cannot produce a fake Merkle tree proof fooling VerifyMTP to accept \( m'_i \neq m_i \in m \) except with negligible probability given \( m \), MT and security parameters. For a consortium blockchain, its advantage lies in allowing efficient comparison and verification of transactions with viable computational power.

**Algorithm 1 BuildMT Algorithm**

1. **Input:** \( m = (m_1, \ldots, m_n) \)
2. **Output:** Merkle tree MT with root
3. **if** \(|m| = 1\) **then**
4. \( \text{label}(\text{root}) = H(m_1) \)
5. **else**
6. \( lchild = \text{BuildMT}(m_1, \ldots, m_{\lfloor n/2 \rfloor}) \)
7. \( rchild = \text{BuildMT}(m_{\lfloor n/2 \rfloor +1}, \ldots, m_n) \)
8. \( \text{label}(\text{root}) = H(\text{root}(lchild) || \text{root}(rchild)) \)
9. **end if**
10. **return** Merkle tree MT with root
Algorithm 2 GenMTP Algorithm

1: **Input**: $MT, \mathcal{H}(m_i)$
2: **Output**: Merkle tree proof $\pi_i$
3: **while** $\mathcal{H}(m_i) \neq \text{label}(\text{root}(MT))$ **do**
4: $lchild \leftarrow \mathcal{H}(m_i).\text{parent}.lchild$
5: $rchild \leftarrow \mathcal{H}(m_i).\text{parent}.rchild$
6: **if** $\mathcal{H}(m_i) = lchild$ **then**
7: $b_j \leftarrow 0, l_j = \text{label}(rchild)$
8: **else**
9: $b_j \leftarrow 1, l_j = \text{label}(lchild)$
10: **end if**
11: $\mathcal{H}(m_i) \leftarrow \mathcal{H}(m_i).\text{parent}$
12: **end while**
13: **return** $\pi_i = ((l_j, b_j))_{j \in [1,n]}

Algorithm 3 VerifyMTP Algorithm

1: **Input**: $\text{root}(MT), \pi_i, \mathcal{H}(m_i)$
2: **Output**: true or false
3: parse $\pi_i$ as a list $((l_j, b_j))_{j \in [1,n]}$, where $l_j$ is a node label, $b_j$ is a binary bit
4: **for** $j$ in $[n]$ **do**
5: **if** $b_j = 0$ **then**
6: $\mathcal{H}(m_i) \leftarrow \mathcal{H}(\mathcal{H}(m_i) || l_j)$
7: **else**
8: $\mathcal{H}(m_i) \leftarrow \mathcal{H}(l_j || \mathcal{H}(m_i))$
9: **end if**
10: **end for**
11: **if** $\mathcal{H}(m_i) \neq \text{label}(\text{root})$ **then**
12: **return** false
13: **else**
14: **return** true
15: **end if**

2.5.2. Consensus Mechanism

The consensus mechanism lies at the heart of a consortium blockchain network, ensuring that all the peer nodes reach the same state. Typically, a consensus mechanism satisfies three key properties: (i) termination, each peer node locally outputs the result within a limited amount of time; (ii) agreement, all honest peer nodes agree on the same value; (iii) safety, the agreed value for all honest peer nodes comes from an honest node. There are multiple perspectives to categorize different consensus mechanisms, e.g., crash fault tolerance (CFT) vs. byzantine fault tolerance (BFT), and different communication models, e.g., synchronous, partial synchronous and asynchronous. We direct readers to related surveys in Table 3 and compare several commonly used consensus mechanisms for consortium blockchains in Table 4.

Table 3. Related surveys involving consensus mechanisms (Fully: ✓, Partial: *, Not Applicable: ×).

<table>
<thead>
<tr>
<th>Surveys</th>
<th>Consensus Comparison</th>
<th>Consortium Blockchain-Specific</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Du et al. [118]</td>
<td>✓</td>
<td>*</td>
<td>2017</td>
</tr>
<tr>
<td>Nguyen et al. [119]</td>
<td>✓</td>
<td>*</td>
<td>2018</td>
</tr>
<tr>
<td>Alsunaidi et al. [120]</td>
<td>✓</td>
<td>*</td>
<td>2020</td>
</tr>
<tr>
<td>Fu et al. [121]</td>
<td>✓</td>
<td>*</td>
<td>2020</td>
</tr>
</tbody>
</table>
Table 3. Cont.

<table>
<thead>
<tr>
<th>Surveys</th>
<th>Consensus Comparison</th>
<th>Consortium Blockchain-Specific</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wan et al. [122]</td>
<td>✓</td>
<td>*</td>
<td>2020</td>
</tr>
<tr>
<td>Ferdous et al. [123]</td>
<td>✓</td>
<td>x</td>
<td>2021</td>
</tr>
<tr>
<td>Lashkari et al. [124]</td>
<td>✓</td>
<td>✓</td>
<td>2021</td>
</tr>
<tr>
<td>Bouraga [125]</td>
<td>✓</td>
<td>*</td>
<td>2021</td>
</tr>
<tr>
<td>Divya et al. [126]</td>
<td>✓</td>
<td>*</td>
<td>2021</td>
</tr>
<tr>
<td>Khan et al. [127]</td>
<td>✓</td>
<td>*</td>
<td>2022</td>
</tr>
<tr>
<td>Yao et al. [128]</td>
<td>✓</td>
<td>✓</td>
<td>2023</td>
</tr>
<tr>
<td>Guru et al. [129]</td>
<td>×</td>
<td>x</td>
<td>2023</td>
</tr>
<tr>
<td>Morais et al. [130]</td>
<td>✓</td>
<td>*</td>
<td>2023</td>
</tr>
</tbody>
</table>

Table 4. Consensus comparison (✩: High; ♦: Mid; ●: Low; →: Transition through timeline; n: the number of nodes in blockchain network).

<table>
<thead>
<tr>
<th>Consensus Mechanisms</th>
<th>Supportive Blockchain Type</th>
<th>Safety</th>
<th>Scalability</th>
<th>Costs</th>
<th>Tolerance Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raft [130,131]</td>
<td>Consortium</td>
<td>♦</td>
<td>♦</td>
<td>●</td>
<td>n/2</td>
</tr>
<tr>
<td>PoS [132]</td>
<td>Consortium, Public</td>
<td>○</td>
<td>○</td>
<td>○→●</td>
<td>n/2</td>
</tr>
<tr>
<td>PoA [133]</td>
<td>Consortium, Private</td>
<td>●→○</td>
<td>●</td>
<td>●</td>
<td>-</td>
</tr>
<tr>
<td>PoET [132]</td>
<td>Consortium, Private</td>
<td>●</td>
<td>○</td>
<td>●</td>
<td>n/2</td>
</tr>
<tr>
<td>PoC [134]</td>
<td>Consortium, Public</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>-</td>
</tr>
<tr>
<td>DPoS [135]</td>
<td>Consortium, Public</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/2</td>
</tr>
<tr>
<td>FBA [136]</td>
<td>Consortium, Private</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>-</td>
</tr>
<tr>
<td>PBFT [32]</td>
<td>Consortium</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>n/3</td>
</tr>
<tr>
<td>RBFT [136]</td>
<td>Consortium</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>n/3</td>
</tr>
<tr>
<td>BFT-SMART [136]</td>
<td>Consortium</td>
<td>○</td>
<td>●</td>
<td>○→●</td>
<td>n/3</td>
</tr>
<tr>
<td>RPCA [136]</td>
<td>Consortium</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/5</td>
</tr>
<tr>
<td>SCP [136]</td>
<td>Consortium, Public</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/3</td>
</tr>
<tr>
<td>HotStuff [136]</td>
<td>Consortium, Private</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/3</td>
</tr>
<tr>
<td>Tendermint [137]</td>
<td>Consortium, Public</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/2</td>
</tr>
<tr>
<td>HoneyBadger [138]</td>
<td>Consortium</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/2</td>
</tr>
<tr>
<td>Dumbo [139]</td>
<td>Public, Consortium, Private</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>n/3</td>
</tr>
</tbody>
</table>

2.5.3. Smart Contract

The term smart contract was popularized by Szabo in a 1994 essay [140]. In consortium blockchain, the smart contract refers to a piece of program that is pre-determined by involved parties, and the logic would be automatically executed without any interruption. When designing a consortium blockchain-enabled application, the Turing complete smart contract can be defined as a stateful ideal functionality [16], i.e., a stateful program that can transparently handle pre-specified functionalities and access the cryptocurrency ledger to faithfully tackle conditional payments once a certain event is triggered.

2.6. Layer II Protocols for Scalability

A consortium blockchain, as a distributed system, faces the critical issue of scalability. To overcome such an issue, a bevy of works focus on the layer-I, i.e., the blockchain
itself, via designing alternative consensus mechanisms [141] or adopting techniques such as sharding [142] and side-chains [143]. However, these layer-I solutions possess shortcomings, e.g., adopting new consensus means changing the core part of a blockchain network, leading to blockchain forking and making the blockchain system backward incompatible. Similarly, sharding implies significant changes in existing blockchain architecture, which seriously hinders its usage in practice.

Layer II protocols are meant to increase the scalability of the underlying blockchain network, thus considerably improving the performance without modifying anything in layer-I. The rationale behind the Layer-II protocols lies in enabling users to perform transactions off-chain via private and authenticated communication. Particularly, layer-II protocols can be divided into three categories: (i) channels, which establish a private p2p medium governed by pre-agreed rules that deployed as smart contracts, and allow users to consent to state updates with state transitions exchanged off-chain [109]. Channels can further be categorized as state channel [144] and payment channel [145,146], where the former is a generalized version and the latter is specific to payment-oriented applications; (ii) commit chains, where an operator can launch a commit-chain and users can join by contacting the operator and submitting transactions. The operator can then periodically submit a commitment to all collected transactions to the parent chain. The typical protocols include NOCUST [147] and Plasma [148]; (iii) protocols for refereed delegation, which function to tackle the disputes among participants, and typically include Truebit [149] and Arbitrum [150].

2.7. Performance Modeling for Consortium Blockchain

Improving the performance of consortium blockchain is undoubtedly of utmost importance. To this end, performance evaluation of the system by experiments is necessary. However, such a process is tedious and time-consuming [151]. It is, therefore, desired to design a model that can compute performance metrics as a function of various consortium blockchain system configurations and parameters. It would facilitate the comparison of different configurations and make design trade-off decisions and meanwhile, enabling users to compute performance for potential architectural updates that the software engineers can take into account for future releases. The existing modeling methods, as tabulated in Table 5 for consortium blockchain can be categorized into the following directions:

- **Queuing.** For consortium blockchain, processes like node competition for consensus transaction confirmation and block generation introduce potential issues such as transaction backlogs and congestion, resulting in increased delays and reduced throughput. Queuing theory can establish models considering interactions among nodes, block propagation times, and transaction confirmations, thus facilitating predicting system performance limits and identifying potential optimizations. The works [152–154] leverage queuing theory to model the different stages of Fabric and Ethereum.

- **Stochastic Petri Nets (SPNs).** SPNs offer a graphical representation that can effectively model the intricate interactions among peer nodes in the consortium blockchain network. Also, SPNs are adept at handling randomness and uncertainty, e.g., network latency, and transaction confirmation time. This stochastic capability is particularly powerful, which enables the analysis of blockchain system performance and stability under varying conditions, thereby facilitating system design optimization and resource utilization. The related works include [151,155,156].

Besides, there are other analytical models proposed for analyzing blockchain performance. Papadis et al. [157] propose a stochastic network model to capture the blockchain dynamics and mainly analyze the impact of the block dissemination delay and hashing power of the member nodes on blockchain performance. Li et al. [158] consider the information propagation delays in the blockchain network and propose Athena, a Hyperledger Fabric-based tuning system that can automatically provide parameter configurations for optimal performance.
Table 5. Different performance modeling methods of consortium blockchain.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Literature</th>
<th>Consensus</th>
<th>Platform</th>
<th>Model Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queuing</td>
<td>[152]</td>
<td>Solo, PBFT</td>
<td>Hyperledger V1.4</td>
<td>Latency</td>
</tr>
<tr>
<td></td>
<td>[153]</td>
<td>Kafka, Raft</td>
<td>Hyperledger V1.4</td>
<td>Throughput and latency</td>
</tr>
<tr>
<td></td>
<td>[154]</td>
<td>POS</td>
<td>Ethereum</td>
<td>Throughput and memory-pool count</td>
</tr>
<tr>
<td></td>
<td>[155]</td>
<td>PBFT</td>
<td>Hyperledger V1.2</td>
<td>Throughput and latency for each phase</td>
</tr>
<tr>
<td></td>
<td>[156]</td>
<td>PBFT</td>
<td>Hyperledger V1.0</td>
<td>Throughput, utilization and mean queue len</td>
</tr>
<tr>
<td></td>
<td>[151]</td>
<td>PBFT</td>
<td>Hyperledger V0.6</td>
<td>Mean Time for Consensus</td>
</tr>
<tr>
<td>Others</td>
<td>[157]</td>
<td>POW</td>
<td>Ethereum</td>
<td>TX processing rate</td>
</tr>
<tr>
<td></td>
<td>[158]</td>
<td>Raft</td>
<td>Hyperledger V1.4</td>
<td>TX throughput and latency</td>
</tr>
</tbody>
</table>

3. Decentralized Applications Atop Consortium Blockchain

As mentioned earlier, the consortium blockchain essentially provides two vital functionalities, i.e., robust storage and guaranteed computing. Hereunder, we describe several popular application scenarios that consortium blockchain empowers.

3.1. Internet of Things

Spawned from the machine-to-machine (M2M) technology, the Internet of Things (IoT) emerges as a new dynamic global network infrastructure with self-configuring capabilities where physical and virtual “things” with identities, physical attributes, and virtual personalities are seamlessly integrated into the information network [7]. According to [159], the number of connected devices will reach 75 billion by the end of 2025. However, such a huge amount of connected devices makes it challenging in terms of the huge amount of collected data, intensive data exchange, security, privacy, centralized processing, and interoperability [160]. To mitigate these issues, (consortium) blockchain has been reckoned as a promising infrastructure. Specifically, (consortium) blockchain along with smart contracts brings advantages in the following aspects: (i) reducing costs, a consortium blockchain can replace the traditional multiple centralized parties charging a lot due to their services, thus reducing the intermediate costs [161]; (ii) establishing trust, a group of parties of interest can join together and establish a consortium blockchain, which enables the transparency and accountability of occurred activities, and therefore, builds trust among participants [162,163]; (iii) privacy protection, blockchain-based public key infrastructure (PKI) allows devices and users to conceal their physical-world identities. By combining with other blockchain-enabled technologies such as decentralized identity [164], privacy can be preserved [165]; (iv) secure information management, consortium blockchain-enabled architecture can realize secure data management by, e.g., issuing certificates for devices [166] and robust information storage [8].

3.2. Healthcare

With the wide adoption of the Internet of Medical Things (IoMT), a great deal of personal health data are collected as personal health records (PHRs). The global PHRs market was about 26.8 billion dollars in 2020 with a probable compound annual growth rate to reach 3.7% by 2028 [167]. Correspondingly, the sharing of PHRs becomes an urgent demand since it can help significantly improve the accuracy of diagnosis, and also be beneficial to disease study [168]. However, PHRs sharing is challenging since the data might be manipulated improperly or revealed during the operational process. Thus, protecting the data integrity and confidentiality is a basic requirement. To address these issues, consortium blockchain emerges as a promising way to build trust among involved hospitals and patients. In particular, [167] proposes a security-aware and privacy-preserved PHR management and sharing scheme based on consortium blockchain where IPFS is
involved to store PHR ciphertext, and zero-knowledge proof (ZKP) to provide evidence for verifying keyword index authentication on-chain. The works [168–170] mainly focus on privacy-preserved data sharing and access control based on consortium blockchain.

3.3. Supply Chain

Supply chain [171] is one of the most straightforward application scenarios of blockchain due to the well-known immutability feature that blockchain can provide, e.g., the IBM Food Trust project built on Hyperledger Fabric [172]. There are many advantages when incorporating blockchain with a supply chain ecosystem. For instance, supply chain management (SCM) involves the design, planning, and execution of all activities that result in the delivery of a product or service to the end customer; a consortium blockchain can increase the effectiveness and efficiencies of global supply chains by delivering relevant information quickly, securely, and efficiently to all participants in the chain and by facilitating the use of digital tokens to track goods as they move along the procurement, planning, production, and delivery phases of a supply chain [173]. Meanwhile, the consortium blockchain-based supply chain finance [174,175] application is a decentralized financial solution that aims to improve the reliability and efficiency of supply chain financial transactions by connecting supply chain participants and financial institutions on a trusted shared ledger. The application leverages the features of consortium blockchain technology, i.e., decentralization, tamper-proof transaction records and smart contracts, to provide more transparent, secure and efficient financial services to all parties in the supply chain.

3.4. Agriculture

Current agricultural advancement and reform are calling for new techniques and innovations to create a more transparent and accountable environment in the agriculture sector [176]. The promising answer lies in blockchain, which can meet the diverse demands in the ecosystem of agricultural products. Existing solutions hinging on a centralized management system suffer from several drawbacks: (i) the centralized server is readily hacked, causing damage to data integrity; (ii) the supply chain management of the agricultural products usually relies on centralized servers, resulting in a single point of failure (SPOF); (iii) high costs are involved to either maintain a set of necessary systems or for a third-party helping to verify and monitor the transactions. To this end, [177] propose a consortium blockchain-based agricultural machinery scheduling system, which optimizes the matching function and scheduling algorithm in the smart contract, and improves the scheduling efficiency. The work [178] proposes a food traceability system based on IoT and blockchain for agricultural products. [70] proposes a consortium blockchain-enabled food trading system, which sets permission for different roles in food transactions and helps choose an optimized trading portfolio for buyers. A set of literature [179–181] investigates the usage of consortium blockchain in the agricultural supply chain. Several works [176,182,183] also review the related techniques, security and privacy challenges and potential research directions in consortium blockchain-enabled agricultural applications.

3.5. Smart Grid

The concept of a smart grid represents a new vision of the traditional power grid that aims to integrate green and renewable energy technologies efficiently [184]. By generating electricity on a small, individual scale and selling it to the grid, it ensures the efficient distribution of electricity, the maintenance of low losses and high quality, and the security of electricity supply [185]. However, challenges such as serious security and privacy issues arise in the adoption of smart grid to consume and trade electricity data [186]. Blockchain technology, which offers a promising solution to these issues, facilitates the following aspects: (i) flexible and integral smart grid data aggregation and regulation [187]; (ii) secure data storage and sharing [188]; (iii) the balance between energy pricing and the amount of traded energy for demand response [189]; (iv) transaction immutability for generators and consumers [185]; (v) privacy preservation for trading users [190,191]; (vi) economic evaluation of blockchain-enabled local energy market [192]; (vii) access
control [193]. The key role of (consortium) blockchain is to act as a trusted third party in a smart grid which is scarce in practice. Furthermore, by integrating cryptographic primitives such as digital signature, encryption algorithms and additional designs such as enhanced consensus mechanisms, blockchain-based solutions can achieve the desired design goals such as reliability, efficiency, flexibility and security in smart grid trading.

4. Challenges and Potential Directions

4.1. Challenges of Consortium Blockchain

We have witnessed the wide adoption of consortium blockchain in many practical settings. Several interesting research directions are still worth further exploration. Hereunder, we highlight the following aspects:

- **Balancing decentralization and performance.** Consortium blockchain-based applications can gain the benefits of being more secure and robust. However, it also introduces extra overhead due to its distributed architecture. As depicted in Figure 5 [194], conventional data centers based on centralized servers can efficiently handle operations. However, the single point of failure issue becomes obvious. The permissionless/public blockchain-enabled systems possess the worst performance but the best robustness. In the middle, the consortium blockchain exhibits better robustness and scalability in comparison with the centralized data centers while having better performance than the fully decentralized public blockchain-enabled systems. Such results follow the so-called blockchain’s impossible triangle, i.e., our current technology and understanding are insufficient to ensure decentralization, scalability and security simultaneously. Though various efforts [195–198] have been put to step towards such an ultimate goal, there still needs to be time to reach it; it is desired to consider the concrete demands when designing consortium blockchain-based systems.

- **Consortium blockchain-enabled provably secure protocol designs.** Following the paradigm of modern cryptography [199], it is indispensable to formally argue the security properties of consortium blockchain-based decentralized applications. Specifically, there are three basic principles needed for probably secure protocols, i.e., formal definitions, precise assumptions and rigorous proofs. In addition, in game-based security, we claim a protocol is secure if the adversary’s advantage is at most negligible considering the security parameter. In the simulation-based security, the protocol is secure if the adversary cannot computationally distinguish between the real-world protocol execution and its simulated version of the security experiment in polynomial time. Moreover, another viewpoint for security proof distinguishes the standalone and the universally composable (UC) model [200], which captures the security of multiple concurrent execution or even composition among multiple secure protocols.

![Figure 5. Blockchain’s impossible triangle (left) and the balancing between decentralization and performance (right).](image-url)
4.2. Potential Directions for Consortium Blockchain

For future research directions, there are opportunities in each layer of consortium blockchain that may influence the development of consortium blockchains.

- **TEE-enhanced designs for consortium blockchain.** Combining Trusted Execution Environments (TEEs) with consortium blockchain exhibits great potential in mitigating various security risks and providing significantly improved efficiency [201]. For instance, designing a more efficient consensus mechanism [202] on TEEs instead of wasting time collecting peer nodes’ responses during reaching consensus, augmenting the confidentiality of smart contracts [203] for consortium blockchain, or designing more efficient and secure consortium blockchain-enabled applications based on TEEs [204]. However, the vulnerabilities [98] exposed by TEEs still require further exploration and solutions.

- **Layer-II protocols for the scalability of consortium blockchain.** Layer-II protocols undoubtedly play a vital role to improve the scalability of consortium blockchains. However, some open challenges, e.g., quantifying the specific cost of these protocols to offer more rationality in performing layer-II transactions, the quantification of layer-II protocols’ decentralization similar to layer-I’s decentralization [205], or providing a systematic method to develop security and privacy notions for layer-two protocols, faithfully including their interaction with layer-I, i.e., the consortium blockchain layer [109].

- **Post-quantum consortium blockchain.** The security properties such as transparency, reliability and consistency of consortium blockchains essentially rely on the underlying cryptographic primitives like public-key cryptography and hash functions [206]. However, the quick advancement of quantum computing has exhibited potential while serious security threats for consortium blockchains. To this end, existing consortium blockchains are expected to be post-quantum, quantum-proof, quantum-safe, or quantum-resistant. Though some efforts [207–209] have been witnessed, no widely recognized post-quantum consortium blockchain platforms are found.

- **Practical application-driven designs.** Consortium blockchain can empower the flourishing of diversified decentralized applications, and it is an ongoing topic to develop a killer application in different settings. Thus, it is worth considering the core functionalities of a consortium blockchain and the abstraction of centralized/decentralized applications [16,210]. The potentially interesting goal lies in building a generalized secure and efficient compiler that can seamlessly transmit the existing architectures to a consortium blockchain-based decentralized ones.

5. Conclusions

Consortium blockchain has been widely applied to many practical scenarios such as finance, IoT, cyber security and the metaverse. It provides two core functionalities of secure robust storage and guaranteed computation, thus bringing many advantages, including a more flexible trust model in comparison with traditional infrastructure. We proposed a layered consortium blockchain architecture and surveyed the pertinent technologies in each layer. Furthermore, the challenges in consortium blockchain itself and building consortium blockchain-enabled decentralized applications are discussed, and the potential research directions are also sketched.

**Author Contributions:** Conceptualization, X.C. and S.H.; methodology, X.C., S.H. and Y.Z.; investigation, X.C. and S.H.; writing—original draft preparation, X.C. and S.H.; writing—review and editing, X.C., S.H., Y.Z. and C.Q.W.; supervision, S.H., L.S. and C.Q.W.; funding acquisition, S.H., L.S. and C.Q.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Key R&D Project of China (No. 2023YFB3308400, 2023YFB3308500, and 2023YFB3308600). Songlin He is also supported in part by NSFC (No. 62302403), the Fundamental Research Funds for the Central Universities (No. A0920502052301-186), and the
New Interdisciplinary Cultivation Fund (No. YH15001124322133) with Southwest Jiaotong University, Sichuan, China.


Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- Defi: Decentralized Finance
- IoT: Internet of Things
- UC: Universal Composability
- MPC: Multi-Party Computation
- SMR: State Machine Replication
- CFT: Crash Fault Tolerance
- PBFT: Practical Byzantine Fault Tolerance
- RTGS: Real-Time Gross Settlement
- PoW: Proof of Work
- PoS: Proof of Stake
- EVM: Ethereum Virtual Machine
- DAPP: Decentralization Application
- TEE: Trusted Execution Environment
- P.P.T.: Probabilistic Polynomial Time
- ECDSA: Elliptic Curve Digital Signature Algorithm
- EU-CMA: Unforgeability Under Chosen Message Attack
- DAG: Directed Acyclic Graphs
- SPNs: Stochastic Petri Nets
- M2M: Machine to Machine
- PHRs: Personal Health Records
- IPFS: InterPlanetary File System
- ZKP: Zero-Knowledge Proof
- SCM: Supply Chain Management
- SPOF: Single Point of Failure

References


46. Farahat, I.S.; Aladrousy, W.; Elhoseny, M.; Elmousey, S.; Tolba, A.E. Secure medical blockchain model. Information 2023, 14, 80. [CrossRef]


49. Liu, L.; Liu, X.; Wan, J. Design of updating encryption algorithm for privacy big data based on consortium blockchain technology. J. Math. 2022, 2022, 7138173. [CrossRef]


52. Shafrakino, A. Indexing structures for the PLS blockchain. Cybersecurity 2022, 5, 19. [CrossRef]


137. Han, R.; Shapiro, G.; Gramoli, V.; Xu, X. On the performance of distributed ledgers for internet of things. *Internet Things 2020*, *10*, 100087. [CrossRef]


180. Alladi, T.; Chamola, V.; Rodrigues, J.J.; Kozlov, S.A. Blockchain in smart grids: A review on different use cases. Sensors 2019, 19, 4862. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.