SURVEY

Exploring Blockchains Interoperability: A Systematic Survey

GANG WANG, University of Connecticut, Storrs, CT, United States

QIN WANG, Commonwealth Scientific and Industrial Research Organisation, Canberra, ACT, Australia

SHIPING CHEN, Commonwealth Scientific and Industrial Research Organisation, Canberra, ACT, Australia

Published: 13 July 2023
Online AM: 08 February 2023
Accepted: 30 January 2023
Revised: 07 January 2023
Received: 09 May 2022

Exploring Blockchains Interoperability: A Systematic Survey

GANG WANG, University of Connecticut, United States
QIN WANG and SHIPING CHEN, CSIRO Data61, Australia

The next-generation blockchain ecosystem is expected to integrate both homogeneous and heterogeneous distributed ledgers. These systems require operations across multiple blockchains to enrich advanced functionalities for future applications. However, the development of blockchain interoperability involves much more complexity regarding the variety of underlying architectures. Guaranteeing the properties of ACID (Atomicity, Consistency, Isolation, Durability) across diverse blockchain systems remains challenging. To clear the fog, this article accordingly provides a comprehensive review of the current progress of blockchain interoperability. We explore the general principles and procedures for interoperable blockchain systems to highlight their design commons. Then, we survey practical instances and compare state-of-the-art systems to present their unique features between distinct solutions. Finally, we discuss critical challenges and point out potential research directions. We believe our work can provide an intuitive guideline for newcomers and also promote rapid development in terms of blockchain interoperability.

CCS Concepts: • **Security and privacy** → *Distributed systems security;*

Additional Key Words and Phrases: Blockchain, interoperability, cross-chain

ACM Reference format:
Gang Wang, Qin Wang, and Shiping Chen. 2023. Exploring Blockchains Interoperability: A Systematic Survey. *ACM Comput. Surv.* 55, 13s, Article 290 (July 2023), 38 pages.
https://doi.org/10.1145/3582882

1 INTRODUCTION

Blockchain [85] is a type of **Decentralized Ledger Technology (DLT)** that heavily relies on cryptographic primitives to provide an immutable and verifiable data platform [63]. It allows a group of distrustful participating nodes (or parties) to provide reliable services. The blockchain is generally used as the tamper-evident log to record data as it is operated by independent parties without a central authority. Based on that, blockchain has become a key enabler for implementing distributed ledgers. Furthermore, the great success of cryptocurrencies demonstrates their superiority beyond traditional centralized architectures. Such emerging technological advances have earned tremendous attention from both industrial and academic domains, promising to change all aspects of digital business. It is believed that blockchain will have a profound influence on existing Internet infrastructures and promote the new trend of Web3 [129].

However, widely adopted blockchain applications have different criteria, necessitating distinct blockchain capabilities. Due to the existence of various protocols and standards, on-chain

Authors' addresses: G. Wang, University of Connecticut, 115 North Eagleville Road, U-3225 Storrs, CT 06269, United States; email: g.wang.china86@gmail.com; Q. Wang and S. Chen, CSIRO Data61, Level 5, 13 Garden Street Eveleigh NSW 2015, Australia; emails: {Qin.Wang, Shiping.Chen}@data61.csiro.au.

0360-0300/2023/07-ART290 $15.00
https://doi.org/10.1145/3582882

information *cannot* be exchanged freely across different blockchains. The development of incompatible blockchain technologies has caused significant fragmentation of the research where users have to choose from a set of blockchains for their specified use cases. This leads to isolation in today's blockchain ecosystem. As a result, we can see many distinct blockchain platforms. It is necessary to achieve interoperable blockchains to enable arbitrary exchanges in future infrastructures [88]. Blockchains equipped with the feature of interoperability would provide a *bridge* to perform open asset-exchanging without jeopardizing original functionalities, greatly improving the flexibility.

Interoperability was initially described as the ability of two or more components to work together despite the existence of differences in language, interface, or execution environment [131]. In the context of blockchain, interoperability means connecting multiple blockchains to mutually access information and act on it by changing the state of each other. It attempts to enable secure state transitions across different blockchains, either homogeneous or heterogeneous. Optimally, this would be achieved without compromising the blockchain's promise of decentralization and trustworthiness [102]. Most existing proposals on blockchain interoperability focus on the process of atomic token exchange across blockchains [75]. To achieve this process, token transferring must be executed in a synchronized process among involved blockchains without the help of a centralized entity. However, atomic token swapping protocols are not sufficiently self-inclusive to complete the task related to state transition requested from cross-chain decentralized applications (*DApps*). This is because the *executable* components in DApps involve more complex activities than pure token transfers. Even worse, such processes always require a counterparty (of another blockchain) who is willing to exchange these tokens [104].

Technically, there is no efficient way to fully replicate or duplicate the state of one blockchain to another blockchain [23]. Completing such a process requires certain efficient schemes to perform the verification of information stored on another blockchain without the help of trusted authorities [104]. It not only needs to consider public blockchains but also needs to cooperate with private and consortium blockchains. However, due to security and privacy, private and consortium blockchains may not be willing to share their information [73]. Successful blockchain interoperability requires at least two blockchains to freely exchange information, which is hard to implement. Besides, many practical challenges must be overcome, especially for scalability, when adapting to a large-scale scenario [65]. Thus, the notion of blockchain interoperability is still in the conceptual stage and has had little practice.

Therefore, ***in this article***, we investigate sufficient *in the wild* projects that claim to achieve interoperability. We dive into their operation mechanisms to capture their common design principles. As a result, we classify existing solutions into different categories regarding blockchain interoperability, namely, *chain-based interoperability*, *bridge-based interoperability*, and *dApp-based interoperability*. For each category, we present the state-of-the-art literature works in each category and provide the corresponding discussion. As a systematization of knowledge on blockchain interoperability, we also point out research challenges and research directions, which assist interested readers in further exploration in this area. A short version of contributions is stated as follows:

- ▷ *We investigate plenty of in the wild blockchain interoperability projects.* We point out their building blocks, abstract the protocol operating models, and discuss different architectures.
- ▷ *We provide a simple but comprehensive classification framework.* The framework covers the architectural design, technique type, features, and properties. We apply the framework to investigated literature and give our summaries in detail (cf. Table 4).
- ▷ *We provide an in-depth discussion on both challenges and opportunities.* A very primary result is that, despite the technique of blockchain interoperability being rapidly developed, it is still

Table 1. Existing Studies (Survey/SoK) on Blockchain Interoperability

Project		Methodology				Subcovered				Discussion		
		Literature Review	*Protocol Abstraction or Modeling*	*Classification*	*Comparison*	*Assumption Clarification*	*Building Blocks*	*Operation Mechanism*	*Technique*	*Property*	*Opportunity*	*Challenge*
VB [28]	Chain interoperability	✗	✗	✓	✓	✗	✗	✓	✓	✗	✓	✓
Borkowski et al. [22]	Towards atomic cross-chain token transfers: State-of-the-art and open questions within TAST	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Kim et al. [65]	A survey of scalability solutions on blockchain	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Schulte et al. [104]	Towards blockchain interoperability	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
Borkowski et al. [21]	Cross-blockchain technologies: Review, state-of-the-art, and outlook	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
Qasse et al. [97]	Inter blockchain communication: A survey	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
Siris et al. [109]	Interledger approaches	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗
Bhatia et al. [16]	Interoperability solutions for blockchain	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Singh et al. [108]	Sidechain technologies in blockchain networks: An examination and state-of-the-art review	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗
Nissl et al. [89]	Towards cross-blockchain smart contracts	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
Lafourcade et al. [69]	About blockchain interoperability	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗
Belchior et al. [13]	A survey on blockchain interoperability: Past, present, and future trends	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
Lohachab et al. [76]	Towards interconnected blockchains: A comprehensive review of the role of interoperability among disparate blockchains	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓
Zamyatin et al. [139]	SoK: Communication across distributed ledgers	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓
This work	Exploring blockchains interoperability: A systematic survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✗= Does not provide property; ✓= Hold the property.

in its infancy stage due to the constraints of survivability, practicability, and security. Based on that, we point out potential research directions to pave the way for communities.

Concurrent Studies. Several pieces of literature discuss chain interoperability in general. Vitalik Buterin [28] classified chain interoperability into three primary categories, namely, centralized or multisig notary schemes, sidechains/relays, and hash-locking. The work briefly introduced the operating mechanisms of each technology. Schulte et al. [104] considered the cross-chain invocation and interaction of both token and smart contracts. They discussed the requirements and benefits obtained from interoperable blockchains. Lafourcade et al. [69] defined a formal model to prove that blockchain interoperability is impossible according to the classical definition of a blockchain. Belchior et al. [13] classified blockchain interoperability into three major categories including cryptocurrency-directed interoperability approaches, blockchain engines, and blockchain connectors. They discussed the existing solutions under the procedure-based framework. Lohachab et al. [76] considered variations in blockchain working principles and operating mechanisms. They classified the solutions into different types according to their layers. Zamyatin et al. [139] abstracted a uniform model for cross-chain communications and formally defined the protocol. They clarified the procedures of running an interoperable protocol.

Besides, Kim et al. [65], Borkowski et al. [22], Nissl et al. [89], Borkowski et al. [21], Qasse et al. [97], Siris et al. [109], Bhatia et al. [16], Singh et al. [108] also proposed their investigations towards blockchain interoperability. We summarize concurrent studies as in Table 1. We capture three major aspects, including *methodology*, *content*, and *discussion*, to evaluate each survey study. In comparison, ***this article*** highlights the technique of interoperability solutions, and we further extend the scope of the aforementioned studies, presenting a comprehensive and systematic study of blockchain interoperability as well as proposing a simple taxonomy method from the perspectives of the functional components.

Paper Organization. The rest of this article is organized as follows: Section 2 introduces preliminaries on blockchains, atomic swaps, and ACID properties. Section 3 gives a bird's view of cross-chain techniques. Section 4 details existing solutions on blockchain interoperability for each category. Section 4 presents our discussion. Section 6 presents opportunities provided by blockchain

interoperability. Section 7 and Section 8 discuss critical challenges and potential research directions, respectively. Section 9 concludes this article.

2 BUILDING BLOCK AND PRELIMINARY

This section provides the necessary background, including the basic concepts of blockchain, ACID properties, atomic swap, cross-chain communication, and interoperability.

2.1 Blockchain

Blockchain Basics. Blockchain is a publicly accessible ledger underpinning digital cryptocurrencies like Bitcoin [85]. In a broad sense, blockchain can be roughly explained as an immutable, decentralized, and trusted *ledger* based on **peer-to-peer (P2P)** networks [106]. It is essentially a distributed data structure that supports on-top decentralized applications, as well as functions to record transactions generated within a network. The key idea behind blockchain technology is decentralization, which means blockchain technology does not require any trusted central point or party to control or manage the participating nodes. Instead, all participating nodes (or peers) in a blockchain-enabled network maintain identical copies of its ledger. All correct nodes are responsible to verify and monitor other nodes' behavior and have the ability to create, authenticate, and verify newly generated transactions. This provides a certain level of security and robustness to guarantee operations on the blockchain are processed correctly in a decentralized manner. Also, it provides some benefits compared with centralized solutions, e.g., tamper-resistance and freedom from the vulnerabilities of single-point failure.

Operation Mechanism. To understand the potential applications of blockchain, it is important to gain a basic understanding of the working principles of blockchain and how it achieves the claimed decentralization. As more transactions are executed and appended, the blockchain ledger continuously grows. When a new block is generated by a certain participating node (e.g., depending on the specified consensus protocol), it must go through a validation process by all other nodes. Once the proposed block is validated by the majority of honest nodes, that block is automatically appended to the end of the blockchain via the inverse reference pointing to its immediately previous block. The first block of a blockchain is called the *genesis* block, and it has no previous blocks. The blocks over the blockchain network achieve a distributed and decentralized synchronization via a *consensus* protocol, which enforces strict rules and common agreements among the participating nodes. Because the blockchain is distributed throughout the whole network, any tampering behavior can be easily detected by other nodes of the network.

Types of Blockchains. Depending on how to organize participants in different scenarios, blockchain can be roughly categorized into three categories, namely, public (or permissionless), private (or permissioned), and consortium (or federated) blockchain [5]. Each category is with distinct attributes, which will further affect the level of interoperability (cf. Table 2).

A *public blockchain*, also known as the *permissionless* blockchain, is an open and transparent network, which implies that anyone can join in the consensus process to construct and verify blocks. It functions in a completely decentralized way. Anyone can maintain an exact copy of the block data and perform the validation process on generated blocks. Typically, this type of blockchain is adopted by most cryptocurrency cases, such as Bitcoin [85] and Ethereum [134]. A public blockchain is designed to support a huge number of anonymous participants, so minimizing potential malicious activities is essential. Due to the anonymous participating process, several types of "proofs" are needed to show the validity of new blocks before publishing them in a public blockchain. For example, proof could be represented as solving a computationally intensive puzzle (PoW) or staking one's cryptocurrency (PoS). Meanwhile, public blockchain normally requires an

Table 2. High-level Comparison of Public, Private, and Consortium Blockchains

	Public	*Private*	*Consortium*
Participants	All	Single organization	Multiple organizations
Identities	Pseudo-anonymous	Approved participants	Approved participants
Permissionless	Yes	No	No
Accessibility	Public Read/Write	Restricted	Restricted
Immutability	Yes	Partial	Partial
Performance	Slow	Fast	Fast
Application Scales	Large	Small	Medium
Major Concern	Accessibility	Privacy	Collaboration

incentive mechanism to reward the peer nodes (e.g., attaching a processing fee on each submitted transaction) for consensus stability. Notably, a public blockchain can prevent itself from being compromised by the incentive mechanism, as it would be too costly to manipulate the contents when thousands of peers are engaged in the same decentralization consensus.

A *private blockchain* is an invitation-only network managed by a central authority.[1] All participants in this blockchain must be granted permission through an authentication mechanism to publish or issue transactions. Any node joining a private blockchain is an authorized member to be agreed upon by the committee. Typically, a private blockchain is used as a distributed synchronized database designed to track information transferring between different sectors. In particular, private blockchain does not need an incentive mechanism, and thus, transaction fees are not needed.

A *consortium blockchain* is similar to the settings on a private blockchain that requires permission to access the blockchain network. Consortium blockchains cover multiple organizations and guarantee consistency and transparency. A consortium blockchain can be considered a verifiable and reliable communication media, which is used to trace synchronized information among its participating members. The accessibility of consortium blockchain lies between the public and private blockchains, which makes it prevalent in the multi-organization involved projects and large-scale industrial systems [40]. In some sense, a consortium blockchain is still one blockchain whose collaborations are within one blockchain. This is different from the concept of interoperability, because these chains are heterogeneous with a high probability.

Different types of blockchains affect the level of interoperability. Before engaging in interoperable operations, they may need extra pre-processing processes. For example, a private blockchain must preserve sensitive information before exchanging information with other blockchains. This will in turn affect the level of interoperability among blockchains.

2.2 ACID of Blockchain

Atomicity, consistency, isolation, and durability (ACID) provide general principles in **database management systems (DBMS)**, with the aim to guarantee the reliability and consistency of a given database [53]. A transaction, in the context of blockchain, is an instance of information exchange. Transactions in an ACID system should hold the following features [114]: (a) a

[1]This central authority does not participate in blockchain construction, and it mainly provides identification-related services.

transaction (or a transaction block consisting of multiple transactions) is executed as a whole or not at all (e.g., enabling the feature of *all or nothing*); (b) each transaction transforms the system from one consistent state to another, without compromising any validation rules or data integrity constraints; (c) concurrent transactions are executed securely and independently, preventing from being affected by other transactions; and (d) once a transaction has been successfully executed, all changes generated by it become permanent even in the case of subsequent failures. ACID is crucial to blockchain transactions as well as cross-chain transactions.

The work [143] proposes two distributed commit protocols, whose approaches enable non-blocking distributed commits for multi-party cross-chain transactions. Both protocols assume that participating blockchains either have an effective way to communicate via smart contracts or a proxy to enable communication. The first one is called a synchronous cross-chain transactions protocol, which follows a **two-phase commit protocol (2PC)** and ACID properties [94], but inevitably results in higher latency. It delays the global commit until none of the participating blockchains can unilaterally roll back the transaction. The second protocol is called the redo-log-based blockchain protocol, and it omits the waiting time before committing a message. However, it relies on a redo mechanism to preserve the system consistency [66].

Besides ACID properties, a multi-blockchain system should follow a SALT property [135]. We have different perspectives regarding the SALT property. From the transaction perspective, a blockchain-based *transaction* can be labeled as Sequential, Agreed, Ledgered, and Tamper-resistant. From the system perspective, a blockchain-based *system* supporting these kinds of transactions can be labeled as Symmetric, Admin-free, Ledgered, and Time-consensual [114]. All these features are key to successfully designing interoperable blockchain systems.

2.3 Atomic Swap

Interoperability requires that individual blockchain systems can communicate with each other, with the ability to share, access, and exchange information across different blockchain networks without an intermediary (e.g., a centralized authority). The information exchanged also requires an atomic swapping process, which can guarantee integrity among different blockchain networks. Technically, the term "atomic" comes from the domain of database systems, in which the execution result of an atomic transaction is confined to a binary value (e.g., either 0 or 1) [84]. In atomic swaps, two parties trade their assets from different blockchains with each other. Both parties need to have an account or an address on the other blockchain, and the trades must happen simultaneously on both blockchains [56]. Both transfers must be guaranteed to happen or neither of them happens. This property is called "atomic," as the swap process is indivisible [52]. One reason that cross-chain swaps are well-known to the blockchain community is that it extends the usability and collaboration among blockchain users. Also, with the help of *smart contracts* [120], the whole swapping process can be executed automatically without human intervention. We can simply consider a smart contract as a script published on the blockchain that establishes and enforces conditions necessary to conduct a transaction, e.g., the asset transferring from one party to another.

Depending on where the transaction happened, atomic swaps can be classified into two major types [84]: *on-chain* atomic swap and *off-chain* atomic swap. In general, an on-chain atomic swapping process happens if an atomic cross-chain swap is between two distinct but homogeneous blockchain networks. In this case, the swapping process can directly be performed on both blockchains. An off-chain atomic swap, however, takes place on a separate layer away from the chains, which can support the swapping process even across heterogeneous blockchain systems. In this case, it requires a "middleware" to facilitate the swapping process.

To implement atomic cross-chain transactions, one way is to make use of **Hash Time-Locked Contracts (HTLC)** [119]. HTLC contracts utilize time locking and pre-image revelation. They

allow a party A (i.e., sending party) to first lock the assets on-chain where the asset can be unlocked in two manners: by party A after a period of time δ, or by a party B (i.e., receiving party) right away only when party B is able to provide proof of execution. By setting two similar contracts on both blockchains, party A and party B can safely exchange their assets without a pre-negotiation or any trusted third party [144]. Another way to achieve atomicity is with the help of a custodian-trusted third party, e.g., a notary scheme or a centralized exchange platform. Both methods should follow the principle of *all-or-nothing* [138].

However, the technical advances of the atomic cross-chain swap still need to overcome many obstacles before being effectively implemented in multi-blockchain systems. In general, the atomic swapping process, especially in on-chain scenarios, are slow in speed, significantly affecting the throughput. Meanwhile, atomic swaps require support from smart contracts. If a blockchain system is compatible, then it is difficult to facilitate the atomic swapping process. Moreover, atomic swaps only solve part of the assets exchange problem between two entities. The need for a fully decentralized exchange is not met, and the swap is subject to a single point of failure.

2.4 Cross-chain Communication

Cross-chain communication is one of the major design considerations in current blockchain systems. Currently, each blockchain system operates as an information-isolated island, where they cannot exchange external data [127]. Cross-chain communication refers to the transferring of information between one or more blockchains [139]. It is motivated by two basic requirements commonly found in distributed systems: accessing or exchanging data and functionality that is available in other systems [100]. Cross-chain communication involves two chains: a source chain and a target chain. The source chain typically refers to the chain that initiates the transactions, and the transaction is executed in the target chain [13].

A typical cross-chain communication protocol refers to the procedure in which a pair of chains (including both intra- and inter-blockchain scenarios) interact to achieve a synchronized status among homogeneous chains [13]. An intra-chain scenario can be, for instance, a sharding blockchain, and each chain can be considered an independent chain maintained by a public shard. While an inter-chain scenario consists of several heterogeneous blockchains, e.g., Bitcoin [85] and Ethereum [134], an intra-chain communication protocol can implement the functionalities to interoperate chains within homogeneous blockchains, while an inter-chain communication protocol requires both source and target blockchains to follow the predefined procedure. Both intra-chain and inter-chain communication protocols are important to fulfill blockchain interoperability, and they can be considered as different-level protocols among multiple blockchain systems.

It is difficult to design a cross-chain communication protocol, since different blockchains may employ very different consensus protocols, block sizes, confirmation times, hashing algorithms, or network models. In the literature, there are several theoretical claims on cross-chain communication. According to the well-known cross-chain proof problem [22], it is hard to detect and verify data recorded on one chain by only observing the exchanged information from another chain. This implies that a target blockchain cannot effectively verify the status or existence of certain data on a source blockchain, especially in the case of lacking trustworthiness between them. A trusted third party, either centralized or decentralized, can help the transferring process. This means cross-chain communication is not feasible in practice without the help of a trusted third party [13]. However, involving a trusted third party is against one of the blockchain features: decentralization. A deep exploration of such issues for blockchain interoperability will exist in a long term.

2.5 Blockchain Interoperability Definitions

Although techniques to advance blockchain interoperability are still in their infancy, and no standardization has gotten agreement, we still find some representative descriptions of the definition of blockchain interoperability. Many literatures [13, 55] mention a definition from the **National Institute of Standards and Technology (NIST)** (NIST Draft NISTIR 8202) on blockchain interoperability: "An interoperable blockchain architecture is a composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain is reachable, verifiable and referenceable by another possibly foreign transaction in a semantically compatible manner." To map the blockchain interoperability into Internet infrastructure, Hardjono et al. [55] defined *survivability* for blockchain systems as "the completion (confirmation) of an application-level transaction independent of blockchain systems involved in achieving the completion of the transaction." Also, the authors point out that interoperability is key to survivability. Thus, interoperability is core to the entire value proposition of blockchain technology." Belchior et al. [13] followed the source-target model and provided the definition as "The ability of a source blockchain to change the state of a target blockchain, enabled by inter- or intra-cross-chain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems."

The above definitions emphasize different domains of blockchain interoperability. Here, we provide our understanding: blockchain interoperability is the ability to correctly conduct assets transferring across a composition of homogeneous and heterogeneous blockchain systems without compromising the legacy design philosophy of each blockchain system. Each blockchain system is an isolated island, and interoperability is an add-on feature to interoperate islands. Thus, when adding new features to blockchains, we cannot compromise the blockchain's original features as a decentralized ledger system.

3 CROSS-CHAIN MODELS

In this section, we first identify different types of transfer and provide a general cross-chain model. Based on that, we give the classification and trust model.

3.1 Types of Transferring

Based on different types of assets, cross-chain transferring can be classified into two categories: *cross-chain token transfers* and *cross-chain state transfers* [66, 104]. We discuss their differences.

Token Transfer. Tokens, *a.k.a.*, cryptocurrencies, are bounded only with one type of blockchain. Transferring tokens between distinct blockchains can be a promising research trend to promote communications. The process should be implemented in a decentralized and autonomously synchronized manner and prevent potential attacks such as double-spending and the faking of transactions [25], or the scenarios of tokens being constructed on the target chain without first being explicitly destroyed on its source chain. One possible solution is to enable both source and target blockchains to verify each other so they can obtain a consistent state. However, considering security, privacy, and efficiency, it is impractical to fully replicate the state of one blockchain within another blockchain to finish the verification process without relying on a third party.

Atomic swaps can help to process the cross-chain token transferring process, which allows clients to swap their tokens on different chains [56]. Atomic swaps do not require a token to be transferred from one chain to another by first deleting tokens on the source blockchain and re-constructing the same amount on the target chain. From the users' perspective, the atomic swapping process provides only the exchange of tokens across distinct blockchains, rather than the transfers of these tokens. However, the process always requires a counterparty who is

willing to exchange their tokens. Another way to perform cross-chain token transfer is to resort to a trusted third party, though this counteracts the feature of decentralization. **Deterministic Cross-Blockchain Token Transfers (DeXTT)** protocol [24] provides a scheme to synchronize the token transferring process across an arbitrary number of chains in a decentralized manner. It allows the tokens to remain available to use even if one of the involved chains is disabled or out of service. To achieve this, DeXTT utilizes the concept of intermediaries, called witnesses, to verify transactions to all peer nodes. But this exacerbates the communication complexity of the whole system.

State Transfer. With the trend of applying smart contracts to DApps, smart contract-based cross-chain mechanisms have become prevalent. Different from cross-chain token transferring mechanisms, cross-chain smart contracts target *general* blockchain interoperability, instead of specific cases of multiple blockchain systems [104]. General blockchain interoperability aims to develop a generic communication scheme that enables smart contracts on one blockchain A to communicate with those on another blockchain B [107].

Several solutions working on such designs have been proposed. Jin et al. [60] provide an architecture for enabling interoperability among multiple blockchains. It includes two operational modes: active mode and passive mode. In passive mode, a blockchain keeps on monitoring transactions or events occurring on another blockchain, while, in active mode, a blockchain first sends information to another blockchain positively and then waits for feedback from that blockchain. Each blockchain has to be aware of the other for communication. PolkaDot [133] provides a more generic multi-blockchain framework, which aims to provide a platform for blockchain interoperability managed by a central relay blockchain, which is used to validate transactions taking place on parachains. Parachains are blockchains that target specific applications and purposes. The purpose of relay blockchain is to use a message-passing protocol to allow parachains to communicate with each other (via inter-chain communication) and process transactions in parallel. Cosmos [67] targets generic blockchain interoperability in industry scenarios. Cosmos uses a blockchain, called the hub, to interconnect independent blockchains, called zones. The token can be transferred as packets between zones through an inter-blockchain communication protocol. The Cosmos hub monitors all committed block headers in the other zones, and each zone maintains track of the hub blocks. Each zone utilizes Merkle tree proofs to prove the presence of messages on its blockchains such that the receiving chain may prove the packet received.

3.2 A Generic Cross-chain Protocol

We present a typical cross-chain protocol for general cases. For simplicity, we consider two independent blockchain systems X and Y. We assume that a process (a.k.a. operation) P runs on X and a process Q runs on Y. A process has the ability to affect blockchain states in two exclusive manners: (a) writing a transaction TX to the blockchain (commit) or (b) stopping to interact with the blockchain system (abort). These assumptions follow the cross-chain communication system model in Reference [139]. A generic protocol consists of the following phases:

(1) Setup. The main task of the setup phase is to exchange and parameterize the information of the involved blockchains, assisting to initialize cross-chain communications so the source and target blockchains can know better of each other. Also, it exchanges the corresponding verification and agreement schemes, including the description summary of transactions. For instance, in an exchange of digital assets, the exchanged information should include the asset types, transferred value, time constraints, and any extra agreement between the two parties. In general, the setup phase happens out-of-band between the involved parties.

(2) Pre-commit on X. Once the setup phase has been successfully settled, a publicly verifiable commitment to execute the cross-chain transaction is submitted on the blockchain system

X, e.g., P write the transaction to blockchain X. And this write operation gets consensus among all honest parties of X via the corresponding consensus protocol. Due to different consensus protocols, the transaction must be in a stable state of the chain X.

(3) Verify. The validity of the commitment value on the chain X by P should be verified by Q following the verification scheme. Two possible results are: Commit on the chain Y or Abort.

(4a) Commit on Y. After a successful verification on the chain Y, a publicly verifiable commitment will be published on the chain Y, and finally, the information will be appended in a stable block.

(4b) Abort. If, unfortunately, the verification process fails, or Q fails to complete the execution of the commitment on Y, then the cross-chain protocol performs an abort operation on blockchain X to "revert" the modification to its original state. This reverting operation can be done by another transaction where the chain X resets to the state before the pre-commit occurs.

For practical considerations, different phases may engage in different operations. For example, the commit (*phase2* and *phase4a*) involves a locking operation and an unlocking operation on exchanged assets of chains X and Y according to the outcome of actual protocol executions. The verification phase can be executed under different trust models, which are related to what exactly is being verified (e.g., consensus agreement on a state or state transition of the transaction). The abort phase is an optional phase, which means once a commit is executed, no abort is necessary. Also, the above generic cross-chain protocol can work with a two-phase commit protocol to facilitate the exchange of assets. The *phase2* (pre-commit on X) is a conditional state transition, which can be reverted based on the execution on chain Y. Also, Zamyatin et al. [139] prove the impossibility where a cross-chain protocol cannot achieve a fair exchange without a trusted third party.

3.3 Classification of Cross-chain Protocols

Considering different rules, there exist different types of classification on cross-chain protocols, e.g., the classifications in References [13, 28]. While, based on the design rationale and use cases, the cross-chain protocols can be classified into two categories: *exchange protocol*s and *asset migration protocol*s [139]. Specifically, *exchange protocol*s synchronize the exchange of assets on two blockchains. The protocol requires an atomic swap of two (or more) digital assets, e.g., x on chain X and y on chain Y. This type of protocol, in practice, consists of a two-phase commit mechanism, where the involved participants can explicitly terminate the exchange process if they fail to reach an agreement. The **hashed time-locked contracts (HTLCs)** is a typical instance (cf. Section 4). *Asset migration protocol*s allow moving an asset or object to a different blockchain. If we consider the two-blockchain scenario that moves an asset from a source blockchain to a target blockchain, then an asset migration protocol is achieved by a "write block" to prevent updates on the source blockchain and to create a representation on the target blockchain. Once the asset migration is successful, moved assets can only be operated on the target blockchain, while the source blockchain loses ownership. Typically, crypto-currency-backed assets adopt asset migration protocols.

3.4 Forms of "Trust Model"

The main challenge of achieving interoperability is the trust model [2]. Trust is one of the prerequisites to conducting the interaction between blockchains. It can be roughly classified into two categories: ***trusted third party* (TTP)** and *synchrony* [139].

Trusted Third Party. A TTP can act in the role of a "coordinator" to ensure the correct execution of cross-chain communications. There are different criteria to classify a coordinator such as by (a) custody of assets vs. involvement in blockchain consensus and (b) static vs. dynamic. For custody

of assets, two forms of custody are involved: *custodian* and *escrow*. In general, custodians have unconditional control over the assets and thus can freely release them. In contrast, escrows have conditional control over the assets, having to obey the predefined constraints. Both types of custody are imperfect. Custodians may commit theft, while escrows may fail to take action (i.e., freeze assets). For the involvement in blockchain consensus, we consider different types of coordinators: *consensus-level* coordinators and *external* coordinators, which are based on their participation in consensus. The second criterion (static vs. dynamic) is typically based on how the election scheme selects the coordinator. A static coordinator cannot be changed over time, whereas a dynamic coordinator can dynamically rotate according to the votes from involved participants.

In practice, the coordinators can be implemented in various forms. For instance, external custodians can be considered in the form of committees, where the trust is spread among the committee members, rather than a single external coordinator. Similarly, consensus-level custodians can be considered in the form of a consensus committee (besides the roles of external custodians), responsible for agreeing on the state update. External escrows can be implemented in the form of multi-signature contracts, requiring a group of individual signatures from involved members. The consensus-level escrow can be realized in the form of a smart contract that can be automatically executed, guaranteeing the executed results have been agreed upon by its consensus participants.

Synchrony. Another type of trust model relies on the assumption of synchronous communication among participants and leverages the locking mechanism. We can alternatively call it *lock contracts*, which facilitates asset exchange and implement a two-phase commit [110]. The locks can be in a symmetric form and can be easily created on both involved chains and then released atomically. In general, this type of trust model is based on the assumption of synchrony, mostly in a synchronous network model with a strong guarantee on the message traversals [34].

Furthermore, there are parallel implementations in practice based on different scenarios, e.g., hash locks [105], signature-based locks [44], timelock puzzles [99], and **verifiable delay functions (VDFs)** [19]. Hash locks rely on the property of *preimage resistance*. Signature-based locks remove the assumption that both parties have to embed the same hash function. Both timelock puzzles and VDF are related to "future" activity, in which the solution to the challenges will be released to the public at a predictable and future time. In general, timelock puzzles build upon inherently sequential functions, e.g., predefined operations, while VDF holds features that guarantee the validity of the released result. Besides, other hybrid schemes such as *watchtowers* [62] can act as an interoperable service provider, assisting to realize interoperability.

4 EXISTING SOLUTIONS ON BLOCKCHAIN INTEROPERABILITY

This section presents concrete solutions. We classify the research on blockchain interoperability into three major categories: *chain-based*, *bridge-based*, and *DApp-based* interoperability (cf. Table 3). Each category has one or more sub-categories. We note that each category is not disjointed, and they may overlap with each other.

4.1 Chain-based Interoperability

The chain-based interoperability mainly targets public blockchains. This category uses token swaps, such as crypto-coin swapping, as a medium to exchange information across different blockchains. We classify three types of on-chain interoperability based on their underpinned techniques (align with [13, 28]): *sidechain*, *notary*, and *hash-locking*. We provide details as follows:

The Sidechain Solution. Sidechain is an essential innovation in blockchain, which affects the broader interoperability and scalability A sidechain can further add new properties, such as security and privacy, to the existing blockchains. The initial goal of the sidechain is to extend the

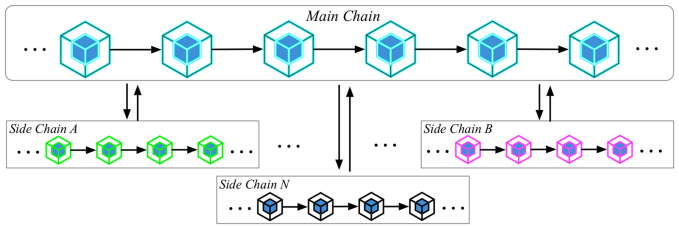

Fig. 1. Abstract of sidechain communication scheme between sidechains and the mainchain.

functionalities of interoperable blockchain networks, where data can be sent and received between the interconnected blockchain networks. This type of design philosophy helps the security of the whole system. For instance, by isolating from the mainchain, in case of cryptographic breaks (or maliciously designed sidechain), the damage is entirely confined to the sidechain itself and will not affect the mainchain. A sidechain enables data to flow between two blockchain systems in a decentralized manner to transfer tokens. Figure 1 shows an abstract model of sidechain communication schemes, with all information going through the main chain.

Focuses on the sidechain should be put on the chain structure and consensus. The mainchain generally does not know the presence of the sidechain. Sidechains must have abilities to locate and follow up the mainchain [38]. Sidechains can have consensus protocols that are completely different from their pegged mainchains. Meanwhile, as a secondary blockchain, the sidechain connects to the main blockchain by leveraging a two-way peg [108]. The heart of any two-way peg lies in a relay routine that transfers data and consensus across blockchains [117].

Two-way Pegs. The initial concept is to design a systematic transfer of assets back and forth between two blockchains. A two-way peg typically operates in the following pattern: A user residing in the mainchain sends their tokens to a dedicated address (a.k.a. *lockbox*) and those tokens are locked on the mainchain. The sidechain receives the locking information on the mainchain and creates the corresponding value of tokens. Then, those tokens can be used by a sidechain user. Meanwhile, the user can transfer tokens back to the mainchain, and the corresponding assets on the sidechain will be either locked or destroyed. An equivalent number of tokens will accordingly be unlocked on the mainchain from the lockbox [13, 108]. We have classified three types of solutions of two-way peg schemes, namely, *centralized two-way pegs*, *federated two-way pegs*, and *simplified payment verification*. Figure 2 shows an overview of these schemes. Figure 2(a) is a centralized scheme where only one central exchange entity manages the transferring process; Figure 2(b) is a federated scheme, which requires multiple entities to collaborate for finishing a transferring process; and Figure 2(c) is an SPV-based scheme, which requires a longer time to complete a transferring process.

Table 3. Classification on Blockchain Interoperability Solutions

Interoperability	***Sub-categories***
Chain-based Interoperability	Sidechain
	Notary Scheme
	Hash-locking
Bridge-based Interoperability	Trusted Relay
	Blockchain Engine
DApp-based Interoperability	Blockchain of Blockchains
	Blockchain Adaptor
	Blockchain Agnostic Protocol

Centralized two-way pegs. This is the simplest way to implement a two-way peg, which requires a trusted third party to manage the locked tokens. The centralized third party is responsible for the operations of locking and unlocking tokens on both the mainchain and pegged sidechains. A centralized scheme provides efficiency, but it is subject to the single point of failure issue.

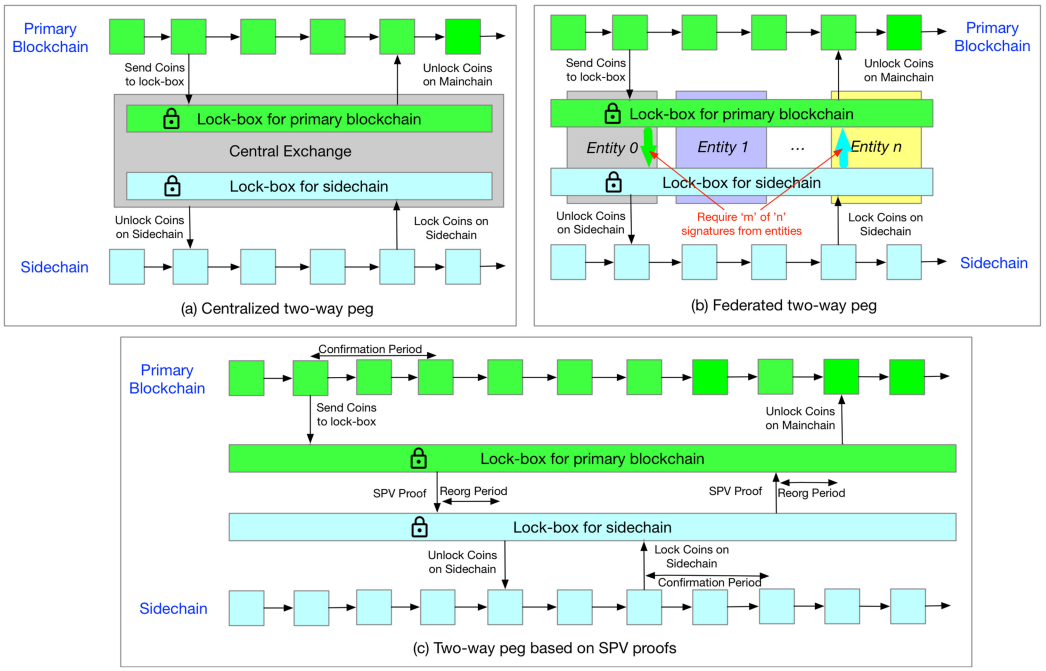

Fig. 2. Two-way peg schemes in (a) centralized, (b) federated, and (c) SPV-based manners.

Multi-signature or federated two-way pegs. Federated two-way pegs are the improved version of centralized two-way pegs, in which a set of notaries control the lockbox, instead of the single entity. In this solution, notaries collectively manage the locking/unlocking operations of tokens between the mainchain and sidechains. The token-transferring process occurs only when the majority of participants reach an agreement. A common implementation is to use multi-signature schemes, in which a quorum of participants signs a transaction. For example, an n out of m ($m >= n$) solution requires at least n participants to sign the transaction for approval.

Simplified Payment Verification (SPV). SPV [64] allows lightweight clients to verify transactions without having to download the full state of the blockchain. Lightweight clients only need to obtain block headers that can significantly reduce the workload. These clients need to request proof information, in the form of a Merkle tree proof [113], to validate whether the target transaction is included in a valid block. A typical SPV two-way peg scheme works as follows: The mainchain tokens are sent to a special address. Only the corresponding sidechain can unlock tokens by showing an SPV proof.

Specifically, the process requires two waiting periods (*confirmation* and *contest* [11]) to synchronize both chains. The confirmation period refers to a period over which a token is locked on the mainchain before its transfer to the sidechain. The period allows for sufficient work to be created. Practically, the length of the confirmation period depends on pre-defined security parameters of the sidechain, which typically requires a tradeoff between speed and security. The contest period refers to a duration in which a newly transferred token may not be used on the sidechain, and the user must wait for this period. The goal is used to prevent attacks, such as double-spending attacks. In this reorganization period, if a user finds any contradictory results to his original request, he can submit Merkle tree proof to show the disagreement. If the submitted proof can indeed prove that it contains a chain with more aggregate work than others, then

this round of conversion will be retroactively invalidated. The process is typically referred to as a reorganization period. The Merkle tree proofs effectively remove the use of third parties.

Advantages vs. Disadvantages. Centralized two-way pegs provide two main advantages: (a1) They are easy to implement and manage due to a simple design, which only involves one centralized entity to control the token transferring process; (a2) The processing speed on token transferring can be extremely fast, as the centralized entity does not need to include complex locking schemes. On the other side, drawbacks come from three aspects: (d1) Centralized schemes go against the design principle of decentralization in blockchains. (d2) The scheme confronts the risk of the single point of failure in multi-blockchain systems. (d3) The centralized entity may steal tokens stored in the lockbox or perform malicious operations if compromised.

Federated two-way pegs provide advantages: (a1) They improve the decentralization of multi-blockchain systems. (a2) They can work with specialized federation protocols (e.g., strong federations [39]) for fast token transfer. In contrast, drawbacks are stated from two aspects: (d1) The design still resorts to a small group of participants to manipulate and monitor token transfer between blockchains, which cannot completely eliminate the centralization problem. (d2) Tokens in the lockbox have risks of being stolen if the majority of the participants are compromised.

The SPV solution provides the advantage of avoiding the use of trusted third parties for token transfer across blockchains. The disadvantage is mainly related to the long time used to complete a transferring process, as a user has to wait for the confirmation and reorganization periods before accessing the transferred tokens on either the mainchain or sidechains.

Platforms of Sidechain. We review four leading sidechain systems, including Loom [77], **RootStock (RSK)** [71], Liquid [39, 87], and **Poof-of-Authority (PoA)** networks [86].

Loom Network is a DApps platform that runs on sidechains, compatible to connect Ethereum [134], Binance Chain [17], and Tron [1]. It is based on a federated two-way peg scheme to swap assets. In its nut, Loom utilizes a **Delegated Proof-of-Stake (DPoS)** protocol [70] for the consensus agreement. Each DApp can independently run atop its sidechain (a DAppChain) and then is pegged to the underlying Ethereum mainchain. Along with DPoS, Loom also runs on a **Byzantine Fault Tolerant (BFT)** consensus [30] as a backend P2P layer (Tendermint [26]). A transaction on the Loom network is not immediately settled on the Ethereum mainchain; instead, it is settled in bulk.

RootStock (RSK) *Network* is a general-purpose smart contract platform where sidechains are pegged to the Bitcoin mainchain. RSK utilizes merged mining [72] to provide incentives to miners who behave actively on the RSK platform. Meanwhile, RSK relies on a combination of a federated two-way peg and an SPV scheme. For the asset transfer, a token of **"SmartBitcoins (SBTC)"** is used to transfer, e.g., from the Bitcoin blockchain to the RSK sidechain. SBTC is essentially a Bitcoin natively on the RSK platform, and this platform can transfer coins back to the Bitcoin network within a specified duration. Each transfer requires a multi-signature to finish the transferring process, where the multi-signature is controlled by the RSK federation. Federation members use hardware security modules to protect their private keys and enforce transaction validation.

Liquid Network is a federated two-way pegged sidechain, relying on the concept of *strong federation*. Originally, strong federations were designed to solve problems of latency, privacy, reliability, and fungibility. A strong federation consists of two independent entities: block signers and watchmen. Block signers maintain the consensus and advance the sidechain, while watchmen realize the cross-chain transactions by signing transactions on the mainchain. Liquid utilizes a multi-signature scheme to sign each block transferred between mainchain and sidechains.

Proof-of-Authority (PoA) *network* is an Ethereum-based sidechain platform. PoA network is intended to allow a cross-chain transferring process between Ethereum to a sidechain with scalability and interoperability. It also provides bridging capabilities that allow users to transfer

their non-fungible tokens from one blockchain to another easily. This feature can be extended for cross-chain smart contracts. PoA network is based on the Proof-of-Authority consensus protocol, where validators can make decisions by themselves independently. PoA network then rewards validators depending on the staked amount. Also, the PoA network provides different types of asset transfers, e.g., Native (i.e., PoA tokens) to ERC 20 [122], ERC20 to ERC20, and ERC20 to Native.

Besides, there exist several ongoing projects to realize blockchain interoperability under a similar design. For example, Plasma [92] aims to provide a highly scalable solution for the blockchain-based decentralized financial industry. Blocknet [36] is a PoS-based platform, which consists of XBridge, XRouter, and iCloud, and XBridges relies on SPV for a two-way pegging process.

Open Issues. The sidechain solutions are still in their infancy stage, coming with several open issues. We put our focus on two major ones as follows:

Centralization. A two-way peg sidechain, either centralized or federated, is subject to the centralization issue. The issue is obvious in centralized two-way pegs. We focus on federated cases. Setting trustworthy individuals as a federation is necessary for this type. However, forming a "good" federation is difficult, even resorting to randomized schemes. A "good" solution has to involve a majority of federation members who are reliable with principles such as verifiable identities of each individual and geographically distributed participants. Meanwhile, the size of a federation should be properly limited. If the size is large, then the verification will spend a much longer time. Otherwise, security might be an issue. Moreover, a federation must rely on consensus protocols to reach an agreement among honest members.

Complexity. SPV-based two-way pegs introduce additional complexity on different levels. On the network level, participants must maintain multiple blockchains to support transfers between each other. It is required that transaction scripts can be rolled back (invalidated) if a reorganization proof comes at a later time. On the asset level, it is no longer a simple assumption of "one chain, one asset." Instead, individual chains may support many assets at the same time. This brings difficulties in the verification process.

Instability. Federated two-way pegs confront the issue of instability, because they require a majority of honest participants to collaboratively sign before passing a transaction. SPV-based designs migrate this issue, however, these solutions are subject to soft-fork due to lack of synchronization among various sidechains [74].

The Notary Solution. The naive idea behind a notary scheme is to introduce a trusted witness to manage the contract ownership or ratification among untrusted parties. A notary is a trusted individual or a group of individuals who can manage multiple chains, initiating transactions in a chain upon the occurrence of valid events or particular requests (e.g., via deployed smart contracts) [13]. The notary is used to claim to one chain that information in another chain is valid by monitoring newly submitted activities and checking their validity. It typically requires a subset of trusted servers [128]. The servers are required to prove the existence and ownership of a given asset within a given time. The immutability property of blockchains allows for the storage of information. It is natural to use the blockchain as a decentralized notary system. Blockchain notary schemes can provide the functionalities of timed proof of existence, whose proof can be further used to prove ownership. Leveraging the notary scheme is a promising solution, because it can facilitate most cross-chain operations and are relatively simple as well as providing a certain degree of decentralization [28]. Figure 3 shows a conceptual communication scheme between two blockchains, and all the information will be recorded by the notary.

Advantages vs. Disadvantages. Notary schemes utilize the third trusted entity as the intermediary between blockchains. One major advantage of the notary scheme is that it is simple, as no additional changes are required in the underlying blockchains. Another advantage is to distribute

trustworthiness to the notaries. They form a trusted committee. The output of these notaries can reach an agreement with the help of the consensus protocols, e.g., **Byzantine Fault Tolerant (BFT)** protocols [29]. Only when two-thirds of the set of notaries [128] have the agreement, the decision can be made.

However, having a notary scheme means that the notary is trusted by users. This means the notary has access to private keys, which makes him vulnerable to attackers. Moreover, the notary controls applications and nodes and has the ability to arbitrarily alter the original transactions. This may lead to the blockchain applications being never trust-free and requiring additional layers to guarantee trust across blockchains. Therefore, while notary schemes take advantage of their atomic process, ease of implementation, and supporting capabilities for different blockchain networks, nevertheless, there still exist main drawbacks such as inefficiency, lack of flexibility, and the risk of centralization. Currently, there is no notary scheme available to handle these drawbacks.

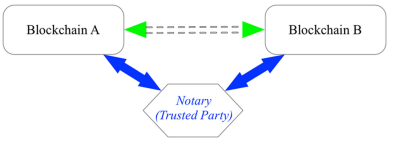

Fig. 3. Abstract of notary scheme via a centralized trusted party as a relay.

Platforms Using Notaries. *Herdius* [12] is a decentralized exchange platform using the notary scheme, with its focus on common connection points between blockchains. The notaries in Herdius are called "assembler nodes," and each one holds a sliced key. It features the solution of using threshold multi-signature schemes. Multiple assemblers can sign a transaction by using the threshold signature scheme. No single assembler can individually decode the native private key without help from other assemblers. *Bifrost* [102] is another project that employs the notary solution, which interacts with multiple blockchains. By using a notary scheme, it is easy to manage data stored on different blockchains without changing the underlying blockchain implementation or maintaining parallel chains. In Bifrost, a user is required to trust its representative notary, which can communicate with other notaries within the system. In general, we can consider the notary scheme adds a trusted layer whose trust depends on the honesty of the notary nodes.

Besides, there exist schemes without explicitly using the notary solution for interoperability. For example, Interledger [118] combines sidechains with notary schemes. AION [112] provides a prototype relying on a notary scheme for an interoperable network. Such projects demonstrate that the combination of notary schemes with other techniques, e.g., the sidechain technique, is a good way to ease the issue of centralization.

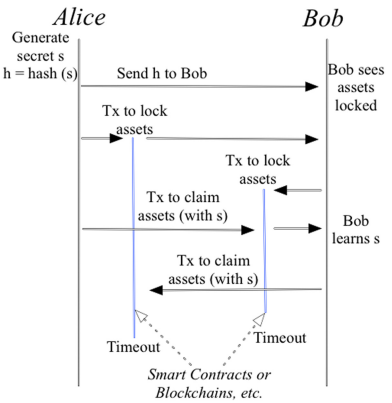

Fig. 4. Abstract of hash-locking communication scheme between two different blockchains [28].

Compared with Sidechains. Notarization is a way to prevent fraud and guarantees that a transaction is genuine that can be trusted. In a notary scheme, the cross-chain transactions highly depend on a third-party notary, and the security relies on the reputation and honesty of involved notaries. Fortunately, most notaries have trusted anchors. These schemes are much like centralized exchanges and banks [98]. Different from sidechains, notary schemes are like a third-party software platform that allows assets to be exchanged among multiple blockchains. In theory, the notary scheme can enable a chain to communicate with arbitrary chains. For example, in cryptocurrency domains, a notary scheme can act as a centralized exchange to allow assets (i.e., various crypto-currencies) to exchange with the guarantee from platform providers.

The Hash-locking Solution. Hash-locking is another technique for the exchange of assets without a trusted third party [91]. The hash-locking technique uses a hash time-locked system, applying a *time lock* to lock transactions. Only when both involved parties are agreed upon obligations, the transaction would keep locked, similar to the concept of an atomic transaction [28, 56]. The exchange is achieved through time difference and hash operation.

Figure 4 shows a hash-locking scheme that performs an atomic swap between Alice and Bob, which connects different blockchains. We assume the sender Alice pays Bob with a lock time t and hash function *hash*. The basic principle of hash-locking works as follows [28]: (1) Alice generates a random secret s and computes a hash value $h = hash(s)$. Then, she sends the hash value h to Bob; (2) Alice and Bob both are required to lock their assets into a smart contract with predefined rules (e.g., Alice locks first, Bob locks after seeing Alice's assets locked). From Alice's perspective, if the secret s is provided within $2t$, then the asset is transferred to Bob, otherwise, the asset is sent back to Alice. From Bob's perspective, if a correct secret is provided within time t, then the asset is transferred to Alice, otherwise, the asset is sent back to Bob; (3) Alice reveals the secret within time t to claim the asset from Bob's contract. The above steps are provably atomic [28].

Another concept in hash-locking is **hashed time-lock contracts (HTLCs)**, enabling cross-chain atomic operations [137]. In an HTLC, a client commits to making the transaction by providing cryptographic proof before a timeout. It is typically used in **Payment Channel Networks (PCNs)** [79]. A payment channel establishes a private peer-to-peer medium, ruled by a set of preset instructions, e.g., smart contracts, which allow the involved participants to consent to state updates by exchanging authenticated state transitions offchain [6, 37]. HTLC is used in PCNs to avoid setting up payment channels, while still preserving high transaction throughput.

Advantages vs. Disadvantages. However, the hash-locking solution is not a one-shot solution. It still has many drawbacks. A hash-locking solution must lock assets during its opening phase for an established transaction channel, but a user confronts the risk of asset loss if the timeout occurs. This may lead to an inconsistent status. Malicious participants can issue many faked transactions to block the normal communication channel, and this will significantly affect other legitimate transactions by honest participants [37].

Platforms by Hash-locking. Besides the use cases in lightning networks, several studies explore the usability of hash-locking for blockchain interoperability.

CheaPay [142] targets a payment channel network, which offers an off-chain settlement of transactions between blockchains. Within a PCN network, it utilizes an HTLC scheme to guarantee the atomic swap of assets. Similarly, ***CHTLC* (Chameleon Hash Time-Lock Contract)** [137] also provides the privacy-preserving service for PCN networks during the asset transferring process. It utilizes the Chameleon-hash function in a multi-layer fashion to guarantee that no user can successfully reclaim the associated payment path unless there exists at least one intermediate payment node (along the payment path) behaving honestly. *Comit* [33] is a protocol stack that facilitates atomic swaps based on HTLCs. It provides two types of protocols, namely, the cryptographic protocol and the communication protocol. The cryptographic protocol defines the order and semantics of interactions with ledgers, while the communication protocol defines the way that two COMIT participants interact to perform an atomic swap. It also provides several specific tokens, e.g., HAN (HTLCs for assets on the ledger), HErc20 (Erc20 asset), and HALight (assets on the lightning ledger), to support direct assets exchange.

***AMHL* (Anonymous multi-hop lock)** [80] defines a cryptographic primitive that functions as a cornerstone for the secure and privacy-preserving PCNs for both scalability and interoperability. It utilizes provably secure cryptographic instantiations to make the AMHLs scheme compatible with current cryptocurrencies. Meanwhile, it utilizes a hash-locking scheme to guarantee atomicity for the token transfer. *Sprites* [83] provides a payment channel to reduce the worst-cast

"collateral cost" for off-chain payments. Its construction relies on a general-purpose primitive, "state channel." To support lined payments, Sprites uses a variation of the standard HTLC, in which a global contract called **PreimageManager (PM)** is created to manage the payment transactions. *Atomic Loans* [18] enables the transfer between various cryptocurrency systems without resorting to a trusted entity. Its atomic swapping scheme is based on an HTLC, and the loan process consists of four phases: the load period, the bidding period, the seizure period, and the refund period. *AltChain* [90] provides a technical prototype of a hash lock for an atomic transferring process. And the prototype is first used by the lighting network in the BTC off-chain transfer expansion solution, which includes the operations of contract locking and unlocked execution and ensures the atomicity of cross-chain transactions [93]. Herlihy et al. [57] construct a complex distributed computing platform to manage cross-chain transactions with a time-lock protocol. The work combines state channels and hashed time-lock contracts to circumvent the scalability limits of existing blockchains.

4.2 Bridge-based Interoperability

Bridge-based interoperability targets the implementation of a *bridge* as a connection component across blockchains. Most existing chain-based blockchains are homogeneous blockchain systems, in which tokens are designed of the same type. For differences, a bridge is to interconnect heterogeneous blockchain systems and cross-chain communication. We categorize them into two major types: *trusted relay* and *blockchain engine*. The bridge sits in the middle of communicated blockchains to maintain the integrity and consistency of involved systems.

The Trusted Relay Solution. Trusted relay is a straightforward approach to facilitate interoperability, where trusted parties redirect transactions from one blockchain to another. Essentially, we can consider a trusted relay as a "bridge" that is used to provide smart contract service between blockchains. A relay enables the recipient chains to verify activities that happened in other chains. Different from notary schemes, trusted relays operate at a chain-to-chain level without posing trustworthiness on distributed nodes. In such a way, relays enable a contract of one chain to work as a "client" of another chain. The scheme replicates block information of the source blockchain via verifiable smart contracts within a target blockchain to allow the target blockchain to verify the existence of data on the source blockchain without requiring trust in a centralized entity [28, 47]. For heterogeneous blockchains, the verification schemes may be very different, and the centralized entity, such as via notary schemes, may be associated with high operational costs. Thus, trusted relays come out in a decentralized way to verify cross-chain communications in a trusted manner.

Platforms of Relay. We briefly introduce several well-known trusted relay-based schemes.

BTC Relay [32] and *PeaceRelay* [78], which utilize SPV scheme to verify transactions across blockchains. These relays are essentially SPV clients for a source blockchain, running on a target blockchain. BTC Relay is a relay built on an Ethereum (target blockchain), which includes transactions that operate on the Bitcoin (source blockchain), requiring authorized clients to submit block headers for verification. Similarly, PeaceRelay is a relay for Ethereum-based blockchains.

Cactus [58] is a part of the Hyperledger project [8], which aims to provide a secure, decentralized, and reliable platform among distinct blockchains. The scheme uses a set of interoperable validators to verify the cross-chain transactions, and these validators are responsible for signing transactions. These transactions must be signed by a quorum of validators to make them valid. Cactus has several transferring patterns: value transfer, value-data transfer, data-value transfer, data transfer, and data merge. Besides, it provides multiple use-case scenarios via a trusted consortium, where trusted relays allow the discovery of the target blockchain. However, Cactus is still not a fully decentralized scheme. Similar to the notary scheme, a set of validators causes the centralization issue.

Testimonium [47] is a blockchain relay scheme that relies on a validation-on-demand pattern and the on-chain execution of SPV to allow verification of data across blockchains without sacrificing the decentralization feature. The scheme consists of relays running on the target blockchain and two types of off-chain clients: one is submitters and the other is disputers. The submitters are responsible for relaying block headers from the source blockchain to the target blockchain, while the disputers are to detect and dispute submitted illegal block headers.

Tesseract [15] is a real-time exchange protocol using trusted hardware as the trusted relay. It supports asset tokenization and can peg these assets to cryptocurrencies. For instance, Tesseract-tokenized bitcoins can be used in the Ethereum chain taking advantage of smart contracts. To achieve it, Tesseract supports cross-chain trading with the help of a **trusted execution environment (TEE)**. The user can establish a secure channel to communicate with the enclave, which provides fast identification and front-running prevention. Also, it enables an atomic cross-chain protocol to achieve *all-or-nothing* settlements.

Besides, there exist other prototype projects on this topic. Kan et al. [61] propose an escrow-based transfer protocol, called interactive multiple blockchain architecture, which is used to exchange information across arbitrary blockchain systems. An inter-blockchain connection model is designed for routing management, and a three-phase commit protocol is used to confirm the communication result. ***SCIP* (Smart Contract Invocation Protocol)** [45] is another protocol prototype that provides a uniform integration for both homogeneous and heterogeneous blockchains. SCIP mainly targets the management of smart contracts, such as supporting methods of triggering smart contract functions, monitoring occurrences of events, and querying past occurrences. A prototype can be implemented at the gateway to coordinate the cross-chain transactions via smart contracts.

Advantages vs. Disadvantages. Trusted relay schemes are highly usable and reliable, with the features of asset portability and atomic swaps and applicable for complex use cases without clear restriction [28]. However, without on-chain validation, centralization issues still exist. Fully decentralized trusted relay networks still have a long way to go. Moreover, performing SPV validation for *every* block header of the source blockchain also leads to extremely high operational costs.

Compared with Notary. Different from notary schemes, relays are operated in a decentralized manner. Newly submitted cross-chain transactions are first verified and validated by the relay, e.g., via smart contract before transactions are transmitted to the target blockchain. The relay needs to know the block information on the source blockchain for successful verification. Existing relays only perform the verification of the source blockchain's header for every submitted block. Then the target blockchain utilizes SPV to verify particular transactions that happened on the source blockchain. The involved blockchains are decoupled through specific procedures, and the relay serves as a bridge between them. The transparent nature of smart technology enables efficient implementation of these relays, while also facilitating decentralization, automation, and accountability.

The Blockchain Engine Solution. A blockchain engine typically requires a shared infrastructure to support services on different layers, including network, consensus, incentive, and so on. The infrastructure provides more than a single role of "relay" among blockchains. Instead, the solution integrates all components for better availability and compatibility. Due to the difficulties in their implementations, most existing blockchain engine-based solutions are still in the stage of proof of concept. Fortunately, we still introduce several projects in progress.

Platforms of Blockchain Engine. We present several leading blockchain engine interoperable systems that as well have outstandingly performed in secondary markets.

Polkadot [27, 133] aims to provide interoperable blockchain networks among heterogeneous multi-chains, which allows the interoperability among many distinct blockchain systems. Beneath

it lies a 'relay chain' that enables simultaneous support of multiple validatable dynamic data structures, known as parachains or parallelized chains, in a side-by-side configuration. Each parachain can be considered an independent chain. In Polkadot, there typically are four basic participants, namely, *validators*, *nominators*, *collators*, and *fishermen*. Validators typically are used to seal new blocks, ensuring the contingency upon enough high-volume bonds being deposited. Nominators play a critical role by acting as a state-holding party who contributes to a security bond on behalf of a validator. The role of collators is to assist validators in producing valid parachain blocks by holding a full status for a particular parachain. Fishermen typically are not required to directly engage in the block-authoring process, functioning as bounty hunters to discover misbehaviors. They also can get rewards for detecting misbehavior and function to ratify invalid parachain blocks. The validator selection is based on a variant PoS scheme for better scalability. For the consensus, Polkadot uses the BFT protocol to get a consensus for newly generated blocks among validators. The validators are then distributed to distinct rotating subsets, one for each parachain to attest to the validity of parachain blocks.

Cosmos [68] is a multi-chain system similar to Polkadot. Each independent parallel blockchain is called a zone (sometimes referred to as a "shard"), which is essentially a Tendermint blockchain [26]. Zones in Cosmos are the blockchains that can plug into the network for data exchange via *Hub*. Hubs connect all of the zones, acting as localized coordinators to ensure that zones communicate in a consistent and standardized manner. In general, the architecture is based on a "hub-and-spoke" structure whereby a set of "spoke" chains link to a "central" hub through its communication protocol, especially, an **Inter-Blockchain Communication (IBC)** protocol. IBC is used to route arbitrary data packets from a source blockchain to a target blockchain, which functions much like the network layer of the Internet Suite. It consists of three layers: (1) bottom - Tendermint, (2) middle - Cosmos network of Zones, and (3) top - Cosmos Hub. The current deployment of Cosmos allows for interoperability among Tendermint blockchains, however, according to its whitepaper, other types of blockchain can also interoperate Cosmos networks, e.g., via peg zones. The cross-chain operation provided by the Cosmos zone, however, highly relies on the feature of instant finality in Hub's state, and some delayed finalization may affect the correctness of these cross-chain operations (e.g., halting the process). In general, Cosmos highly depends on the correct behaviors of validators to offer interoperability, in which it utilizes BFT consensus protocol and peg zones to provide overall consistency and interoperability.

WanChain [124] can support cross-chain operations among chains. It adopts a PoS consensus protocol for the consensus, which was originally forked from an Ethereum-based generic ledger. Wanchain uses both multi-party computing and threshold secret-sharing technologies to perform account management without involving any trusted third party. The cross-chain protocol includes three key modules: the registration module (e.g., registering the original chain participating in cross-chain transactions and registering assets to be transferred), the cross-chain transaction data transmission module (e.g., creating transaction requests and acknowledging the receipt of transactions), and the transaction status query module (e.g., providing the querying service on the confirmation status of involved assets). The verification nodes can be divided into three categories: vouchers (functioning as cross-chain transaction proof nodes), storemen (functioning as locked account management nodes), and validators (functioning as general verification nodes). The project announced the release of the **T-Bridge (Trusted Bridge)** to enable universal blockchain connections. The T-Bridge model connects components from the source chain, target chain, and routing chain. Via smart contracts deployed in blockchains, T-Bridge enables both users and service providers from different blockchains to perform cross-chain operations.

ARK [10] tries to provide a platform for blockchain interoperability, which mostly relies on the concept of the bridge. ARK specializes in building customizable bridge chains that can interlink

Table 4. Blockchain Interoperability Solutions

			Feature						Property		
	Project	Type	*Transfer Type*	*Trust Model*	*Connected Chains*	*Consensus*	*Centralization*	*Technique*	*Stability*	*Scalability*	*Complexity*
Chain-based	Loom Network [77]	Sidechain	Tx	Sychrony	Two	DPoS	Mainchain	-	✓	-	-
	RootStock [71]		Tx	Sychrony	Two	PoW (Bitcoin)	Mainchain	Merge mining	✓	-	-
	Liquid Network [87]		Tx	Sychrony	Two	PoW (Bitcoin)	Mainchain	Multi-sig	-	-	-
	PoA Network [86]		Tx	Sychrony	Two	PoA	Mainchain	-	✓	-	-
	Herdius [12]	Notary		TTP	Two	PoW (Ethereum)	Notary	Threshold sig	✓	-	-
	Bifrost [102]		Tx	TTP	Two	-	Notary	API	✓	-	-
	Interledger [118]		Tx	TTP	Two	PoW	Notary	-	✓	-	-
	AION [112]		Tx	TTP	Two	N/A	Notary	-	-	-	-
	CheaPay [142]	Hash Locking	Tx	Sychrony	Two	-	Mainchain	CHTLC	✓	-	-
	AMHL [80]		Tx	Sychrony	Two	-	Mainchain	Multi-hop lock	✓	-	-
	Sprites [83]		Tx	Sychrony	Two	-	Mainchain	HTLC	✓	-	-
	Atomic Loans [18]		Tx	Sychrony	Two	-	Mainchain	HTLC	✓	-	-
	AltChain [90]		Tx	Sychrony	Two	PoW (Bitcoin)	Mainchain	HTLC	✓	-	-
Bridge-based	BTC Relay [32]	Relay	Tx	TTP	Two	PoW (Bitcoin)	Mainchain	SPV	✓	-	-
	PeaceRelay [78]		Tx	TTP	Two	PoW (Bitcoin)	Mainchain	SPV	✓	-	-
	Cactus [58]		State	TTP	Two	BFT (Hyperledger)	Validator	-	-	-	-
	Testimonium [47]		State	TTP	Two	-	Mainchain	SPV	✓	-	-
	Tesseract [15]		State	TTP	Two	-	Mainchain	TEE	✓	-	-
	Kan's [61]		State	TTP	Two	BFT	Mainchain	-	✓	-	-
	SCIP [45]		State	TTP	Two	PoW (Ethereum)	Mainchain	-	✓	-	-
	Polkadot [27]	BC Engine	State	TTP	Multiple	PoS	Validator	Parachain	✓	✓	-
	Cosmos [68]		State	TTP	Multiple	BFT (Tendermint)	Hub	IBC	✓	✓	-
	Wanchain [124]		State	TTP	Multiple	PoS	Validator	MPC	✓	✓	-
	ARK [10]		State	TTP	Multiple	DPoS	Validator	SmartBridge	✓	✓	-
DApp-based	Overledger [121]	BoB	State	TTP	Multiple	BFT	Validator	Layered design	✓	✓	-
	HyperService [75]		State	TTP	Multiple	-	Mainchain	UIP	✓	-	-
	Fabric [8]		State	TTP	Multiple	BFT (Hyperledger)	Validator	-	✓	✓	-
	SMChain [125]		State	TTP	Multiple	-	Mainchain	Two-layer	✓	-	-
	Block Collider [115]		State	TTP	Multiple	N/A	Mainchain	-	✓	-	-
	CAPER [7]		State	TTP	Multiple	BFT	Validator	DAG	✓	✓	-
	Frauenthaler's [46]	Adapter	State	TTP	Multiple	-	Mainchain	Monitor	✓	-	-
	PleBeuS [103]		State	TTP	Multiple	-	Mainchain	Policy-based	✓	-	-
	Move [48]		State	TTP	Multiple	-	Mainchain	Contract-base	✓	✓	-
	Interledger [118]	Agnostic	Tx	TTP	Multiple	BFT	Validator	-	✓	-	-
	Perun [116]		State	TTP	Multiple	BFT (Hyperledger)	Validator	Layer-2	✓	✓	-
	Gravity [96]		State	TTP	Multiple	-	Mainchain	-	✓	✓	-
	SuSy [95]		State	TTP	Multiple	-	Mainchain	Gateway	✓	✓	-

– = Does not provide property; N/A = Not known due to the absence of supporting documents; **Abbr.**: BC = Blockchain; Tx = Transaction; IBC = Inter-Blockchain Protocol; UIP = Universal Inter-blockchain Protocol; CHTLC = Chameleon Hash Time-Lock Contract; MPC = Multi Party Computation; TTP = Trusted Third Party.

or operate independently, providing users with a seamless way to exchange data. ARK offers interoperability via a multi-chain approach and ARK SmartBridge technology, in which complex processes are executed on the bridge chain and only the execution results will be transferred back to the main chain. ARK's SmartBridge defines two types of communication protocols: Protocol-specific SmartBridge and Protocol-agnostic SmartBridge (or Protocol-independent SmartBridge). The former refers to a communication layer targeting ARK-based application-centric blockchains that operate within the ARK network. The latter aims to connect blockchains that adopt different consensus protocols, which mainly are used for cross-chain communications. The project utilizes **Delegated Proof-of-Stack (DPoS)** as the consensus algorithm to validate transactions. Holders of ARK as the delegates vote on the transactions, insert blockchain, and create new ARKs.

4.3 DApp-based Interoperability

Various applications have been built on top of blockchains, benefiting from their decentralization, immutability, and trustworthiness. We call these types of applications decentralized applications, which are Internet applications operating on a decentralized P2P network (blockchain). However, DApp cannot singly ensure semantic interoperability. Thus, it is essential to ensure that a DApp supports minimum structural interoperability and gradually achieves interoperability among DApps. We discuss three types of approaches to achieve DApp-based blockchain interoperability, namely, *blockchain of blockchains*, *blockchain adapters*, and *blockchain agnostic protocols*.

Blockchain of Blockchains. The **blockchain of blockchains (BoB)** provides a platform for developers to construct cross-chain DApps, and each blockchain can operate independently. For simplicity, we consider the top-level blockchain as the mainchain, and every blockchain in BoB is a participant of the mainchain, functioning as a subchain. Each subchain is a user to access the mainchain internet. Intuitively, it looks like a sidechain solution. However, it is practically different from a sidechain solution. Sidechain solutions run subchains of the mainchain, where all actions should be coordinated by the mainchain. BoB is more like a notary scheme (in implementation), where the mainchain serves as a notary to record activities that happen on heterogeneous subchains.

Platforms of BoB. Due to various application scenarios, we discuss BoB schemes case-by-case.

Overledger [121] abstracts a single-ledger dependent technology to integrate different architectures, as well as introduces a vendor-independent protocol to achieve message-oriented communication. The architecture has four layers: a *transaction layer*, a *messaging layer*, a *filtering and ordering layer*, and an *application layer*. The transaction layer stores transactions appended on the ledger, including all operations in diverse blockchain domains. This layer can operate on different ledgers. The messaging layer is to retrieve and stores relevant information from different ledgers. The information includes transaction data, smart contracts, or metadata. It can be considered a shared channel for packets from different applications. The filtering and ordering layer is in charge of connecting the various messages from the messaging layer. This layer extracts messages from transaction information. It provides a filtering service to filter out unnecessary information and orders them into the block. The validation scheme examines the application scenarios and its specification. Such information can be extracted from transaction data. The application layer is an upper part of the reference architecture, interacting with applications, where messages from different applications may be shared or referred to by other applications. Finally, communications in Overledger adopt a similar two-phase commit protocol scheme for atomic commitments.

HyperService [75] is a platform to offer interoperability and programmability across heterogeneous DApps. It facilitates DApp development by providing a virtualization layer on top of the underlying heterogeneous blockchains, yielding a unified model and a high-level language to describe and program DApps. Users can easily write cross-chain DApps via the provided interfaces. HyperService utilizes a **Universal Inter-blockchain Protocol (UIP)** to handle the complexity of cross-chain execution, which can operate on any blockchain with a public transaction ledger in a secure and atomic manner. In general, UIP can securely execute cross-chain operations that may further involve the execution of smart contracts deployed on heterogeneous blockchains. Also, the UIP is a fully trust-free solution without trusted entities involved. HyperService includes four key components. *DApp clients* essentially functions as the gateways to connect DApps to the HyperService platform, a lightweight client interface. ***Verifiable Execution Systems (VESes)*** act as blockchain drivers, converting high-level DApp programs provided by clients into blockchain-executable transactions. Both VESes and DApp clients employ the underlying UIP protocol, and UIP itself contains another two building blocks: ***Network Status Blockchain (NSB)*** and ***Insurance Smart Contracts (ISCs)***. NSB serves as a BoB to provide an objective and unified view of

DApps' execution status based on the execution of ISCs. ISCs can revert executed transactions to guarantee financial atomicity and offer accountability of entities.

Hyperledger Fabric [8] is a modular and extensible system for distributed applications. It provides support for modular consensus protocols, and this allows the deployed system to be more customized (e.g., targeting specific use cases and different trust models). Fabric introduces a novel *execute-order-validate* blockchain architecture. Practically, a general application deployed on Fabric consists of two components: a chaincode and an endorsement policy. A chaincode essentially is a smart contract that executes the application logic during execution. The chaincode is the most important component for distributed applications, and this code may be provided by an untrusted developer. Also, it has a system chaincode to manage the blockchain system and maintain parameters. During the validation phase, an endorsement policy will be accessed and evaluated.

SMChain [125] adopts a two-layer blockchain structure, consisting of independent local blockchains stored at individual plants and one state blockchain stored in the cloud. In SMChain, each local chain maintains its private ledger, preventing any non-member from modifying it at any time. Different local chains may run different consensus protocols in parallel. There is no asset exchange across local chains, and only the status of each local chain is collected and stored in a state chain. The state chain then builds blocks based on the information of local chains, and these blocks will be returned to local chains for integrity and interoperability checks. By allowing a two-layer structure, it can achieve a certain level of interoperability on the status of local chains.

Besides, there are other works, e.g., *Block Collider* [115] and *CAPER* [7]. Block Collider aims to be a multi-chain platform. Developers can modularly combine exotic features from blockchain across the multi-chain platform and can build in the capability of load balance between chains. CAPER is a permissioned blockchain platform to support both internal and cross-chain transactions of multiple collaborating DApps. The application-specific blockchain maintains a directed acyclic graph where each application is restricted to access and maintain its ledger.

Blockchain Adapters. A blockchain adapter targets end-users by providing an interface to allow them to interoperability, e.g., via runtime selection or smart contracts.

Platforms of Adapters. We summarize projects that adopt the blockchain adapter method.

Frauenthaler et al. [46] propose a framework for blockchain runtime selection. The proposed solution actively monitors the status of multiple blockchains, which can help users to choose the most appropriate blockchain and provide the switch-over service between blockchains even during the runtime. However, it must continuously monitor multiple blockchains simultaneously. The framework allows for seamless integration with other blockchains, meaning that if a more suitable option becomes available, the system can automatically route future operations to the new chain. Also, user-defined data stored on the current blockchain can be moved to the target chain. In general, the presented framework consists of three key components: the monitoring component, the blockchain selection algorithm, and the switchover component. The monitoring component continuously monitors and calculates metric values. The blockchain selection algorithm, based on the calculated metric values on each blockchain, selects the most beneficial one. And the switchover component provides the ability to switch from one chain to another.

PleBeuS [103] is a policy-based blockchain selection scheme that follows their previous two policy-based selection works [101] and Bifrost [102]. PleBeuS adopts a generic cost-aware method with consideration of both public and private blockchains and their technical specifications. By communicating with a BC-agnostic API, PleBeuS can enforce the interoperability of transactions. PleBeuS follows the concept of **Policy-Based Management (PBM)**, which consists of a **Policy Management Tool (PMT)**, a **Policy Decision Point (PDP)**, and a Bifrost API acting as a **Policy Enforcement Point (PEP)**. PleBeuS implements a cost-aware policy-switching mechanism

and two blockchain selection algorithms. The blockchain selections can be of two types, either performance-targeted or cost-targeted.

Move [48] is a smart contract-based protocol for interoperability. It offers developers an operational primitive, enabling contracts and assets to switch between blockchains, with the guarantee of most key blockchain properties such as consistency. The move protocol divides a move operation into two separate transactions, *Move1* and *Move2*. Move1 is used to lock the smart contract state in the source blockchain, while Move2 is to reconstruct the smart contract on the target blockchain. Similar to two-phase commit protocols, it adds constraints to the sequence. Only if the Move1 transaction has been executed successfully under proofs, the Move2 transaction can proceed.

Blockchain-agnostic Protocols. Blockchain agnosticism refers to a single platform allowing multiple blockchains to co-exist, enabling cross-chain communication between arbitrarily distributed ledgers. In essence, blockchain agnosticism provides its end-users with various options to pick their optimal blockchain and provides the capabilities for migrations between blockchains.

Platforms of Agnostic Protocols. We introduce the projects of blockchain-agnostic protocols.

Autonomous System. A design philosophy for interoperable blockchain systems, or an interoperability architecture for blockchain autonomous systems following the design principle is proposed in Reference [55]. Both are blockchain-agnostic protocols. The proposed framework is based on **autonomous systems (AS)** (alternatively called routing domains) as one type of connectivity unit (like one single participant in blockchain systems) to offer the scale-up capability. The domains of blockchain systems can be considered as a connected set of "islands" of AS, stitched together through peering agreements. Blockchain gateways in AS play a key role to achieve interconnectivity and thus interoperability, in which gateways are used to execute and validate cross-chain transactions. The framework has two types of nodes: one is intradomain nodes, responsible to maintain ledger information and conducting transactions within one domain; the other is interdomain nodes (a.k.a. interdomain gateways), which are used to handle cross-domain transactions involving different blockchain ASs. Each AS domain can be owned by a private organization in the form of private blockchains. It is crucial to guarantee information confidentiality of private blockchains, and the framework also provides a use case for this requirement.

***Interledger Protocol* (ILP)** [118] is designed for a payment network across different payment systems, which provides a way to secure transferring process between ledgers. The core of ILP is a concept of the connector that is used to coordinate the token-transferring process on distinct ledgers. Connectors can also serve as a translator between different ledger protocols. Typically, the atomicity is ensured by a BFT algorithm to guarantee the consistency of the ledger's state. A new version of ILP, called ILPv4 [59], can be adopted into other distributed ledgers, instead of only on the payment network. A participant in ILPv4 has one or more roles: sender, receiver, or connector. A connector is an intermediary between a sender and a receiver that forwards ILP packets. ILPv4 utilizes payment channels for settling bilateral payment obligations, with its packets sent only between connectors, without involving the participation of underlying ledgers. This means the packets communicated in ILPv4 are based on forwarding, instead of delivery, and the connectors forward packets based on their local exchange rates, instead of a fixed destination rate in the version of ILP. ILPv4 consists of three different types of packets: *Prepare*, *Fulfill*, and *Reject*, which roughly correspond to request, response, and error messages in a client-server communication model, respectively. In general, connectors can forward *Prepare* packets to the corresponding receivers, and the connectors transit the *Fulfill* or *Reject* packets back to the representative senders.

Perun [116] is initially a joint DLT layer-2 scaling project, and currently joins the Hyperledger ecosystem as one of the lab projects. Perun is a blockchain-agnostic state channel framework,

aiming to make blockchain ready for mass adoption and alleviate current technical challenges such as high fees, latency, and low transaction throughput. The project is a modular design, enabling the flexible integration of Perun's state-channel technology into any blockchain or traditional ledger system. It allows state-channel virtualization, and virtual channels can be established and closed with the help of state-channel intermediaries. Perun enables interoperability via blockchain-agnostic design and state-channel virtualization, and this further allows smart contracts that can be executed across different blockchains.

Gravity [96] is a blockchain-agnostic cross-chain protocol between blockchains and external entities (e.g., data oracles). The network consists of a non-isomorphic Gravity node. A gravity node consists of the *core* (responsible for all business logic) and data feed extractors (e.g., in the form of boilerplate source code). Providers can openly choose to operate in one or more target chains, or they can implement extractors to extract necessary data. In general, Gravity can be considered a singular decentralized oracle. *SuSy* [95] is a blockchain-agnostic cross-chain gateway protocol based on Gravity, a second layer protocol over Gravity. The current version of SuSy focuses on token transferring without bringing incentive models for transfer providers. To be noted, the protocol highly relies on the trusted oracle model, which acts as an intermediary in the information-transferring process between blockchains.

Besides the main trends mentioned above, there exist some conceptual works on blockchain agnosticism. For example, a framework, called a blockchain router, is proposed for cross-chain communication in Reference [127]. A blockchain router consists of four different participants: validators, nominators, surveillants, and connectors. Each participant has a distinct functionality. For instance, the validators in the blockchain network are responsible to verify, concatenate, and forward blocks to the correct destination. Another example is a framework for inter-blockchain communication [61], which also is a blockchain-agnostic protocol. This framework focuses on the transaction design, which enables heterogeneous blockchains to communicate with each other through standard crossing-chain transactions. And the crossing-chain transactions are transferred by nodes in the router blockchain in a peer-to-peer manner without the participation of any third party.

5 DISCUSSION

In this section, we separately discuss each method to achieve interoperability.

5.1 Towards Chain-based Interoperability

Sidechain solutions cannot only provide interoperability among multiple chains but also increase the scalability of the mainchain by performing pre-processing before submitting transactions to the mainchain. Meanwhile, sidechain-based projects target permissioned blockchains where authorized participants can conduct instance decisions, improving confirmation and scalability. Different organizations can develop their own sidechain (e.g., as an atomic zone) via the mainchain for communications. From this perspective, sidechain solutions provide a type of isolation in nature, and this can potentially protect the entire system. For example, a sidechain may be compromised by an attacker. Due to isolation, a sidechain provides a barrier against the propagation of malicious behaviors in the mainchain when being compromised by an attacker.

However, the sidechain solution is not a perfect solution to deal with interoperability. The first issue in the sidechain is centralization. Centralized or federated two-way pegs in the sidechain solution introduce a certain level of centralization; a federated scheme, for example, is critical to choose trustworthy entities as its members. The integrity and security of these systems highly rely on the honesty of the federation. If a malicious group of entities forms the majority within a federation, then it may introduce a security flaw. For instance, the assets might be locked in the

lock-box, which cannot be redeemed. SPV provides a solution against the centralization issue in federated two-way pegs by using a group of entities for transferring assets between the mainchain and sidechains. However, the solution requires a long verification process to finish a transferring process, as the node needs to wait for confirmation and reorganization periods.

The notary scheme is an intermediary to validate blockchain transactions. Because of the decoupling with underlying blockchains, this solution works for most blockchains and is comparably easy to implement. However, the use of notary schemes weakens the feature of decentralization, since few blockchain applications are completely trustworthy. The inevitable issue of centralization becomes significant. It makes the notary a target to attack, being vulnerable to the single point of failure attack. Hash-locking solutions can allow asset exchange in a trustless way from the chain level. However, since the hash-locking scheme is based on the lock/unlock mechanism, assets may get lost due to the timeout. Also, for each atomic swap, multiple transactions may yield a long waiting time. Protocols on chain-based interoperability are still in infancy and some of them are still in conceptual and prototype design. Generally, a practical way to enable chain-based interoperability with current leading blockchains is to combine these solutions together, e.g., combining the sidechain technique with decentralized forms of a notary scheme.

5.2 Towards Bridge-based Interoperability

Bridge-based interoperability solutions are used to deal with heterogeneous multi-blockchain systems, in which each blockchain has a different chain structure, verification mechanism, consensus protocol, smart contract, and so on. The use of a bridge can be as a channel or connector to get rid of the incompatibilities and makes cross-chain communications manageable. If we compare it with the Internet protocol, then the bridge functions as a router, where the outputs from a source blockchain network are processed and transferred to another blockchain. In general, trusted relays are much simplified that can be easily adopted. The solution utilizes mechanisms similar to the notary schemes, causing a certain degree of centralization. Blockchain engines are very recent solutions, which utilize a similar design to two-way pegged sidechains or hash-locking solutions.

Bridge-based blockchain solutions may adopt different cross-chain protocols. Polkadot uses **cross-chain message passing (XCMP)**, while Cosmos uses **inter-blockchain communication (IBC)** protocols. The cross-chain scheme is highly related to their overall architecture design, such as the roles of nodes. Current protocols are imperfect, and each communication protocol has both advantages and disadvantages. IBC is a more generic solution than XCMP, which can allow users to customize their zones and provide security promises. XCMP restricts these customizations but offers a secure framework for communications via a shared security layer.

The solution provides convenience for end-users, and they do not need to know what happened in the *bridge*. It is much like an **Internet Protocol (IP)** in an IP suite, where end-users only need to send these *packets* (cross-chain transactions) to the bridge. Different from IP protocol, the bridge solution does not interoperate with each other, and they require specific bridges to handle the communication among heterogeneous blockchains. Bridge-based blockchains require transaction fees to keep a fair operation. This may further limit the use of interoperable blockchain. It would be desirable to define a standardized bridge solution via international organizations so different blockchains can work smoothly based on principles agreed upon by others. However, the standardization processes of blockchain interoperability still have a long way to go.

5.3 Towards DApp-based Interoperability

DApp-based blockchain interoperability has great potential to realize blockchain interoperability, even though most of these solutions are still in their infancy. A BoB scheme typically requires a second layer of blockchain to record its sub-blockchains. Different from notary schemes in

chain-based interoperability, this chain functions purely as a *notary* to record the activities among sub-blockchains. Cross-chain communications can happen between heterogeneous blockchains. Blockchain adapter solutions provide flexibility to end-users and let them decide the most appropriate solutions for their blockchains. This category focuses on API design and enables data portability. However, most of the works presented lack a practical implementation with criteria to evaluate their effectiveness and efficiency. As an adapter, some solutions appear somewhat centralized, especially ones that require a direct connection with a trusted party. Blockchain-agnostic solutions are independent, which offers interoperability to existing blockchains. Most solutions in this category focus on prototype design, with more generalization than solutions in the blockchain adapter. That means blockchain-agnostic protocols provide flexibility to the adaptation of the selection of blockchains, and the selection does not rely on the underlying blockchains. However, most schemes in this category cannot support backward compatibility.

6 OPPORTUNITIES

This section focuses on the opportunities provided by blockchain interoperability in various applications and other technologies that can be promoted by blockchain interoperability.

6.1 Blockchain Interoperability Applications

Interoperability provides a way to achieve faster, more efficient, and highly secure business-to-business or business-to-consumer transactions. Most existing network-based applications can benefit from blockchain interoperability, which enables free exchange of information in a trusted, immutable, and decentralized manner. We list typical applications that can benefit from blockchain interoperability, from perspectives of *supply chain*, *healthcare*, and *industry*.

Supply Chain. Supply chain is a general term that can be adopted into various applications, e.g., transportation industry, food supply chains, pharmaceutical supply chains, and manufacturing supply chains. To successfully apply blockchain to supply chain applications, several key challenges need to be resolved, namely, traceability, dispute resolution, cargo integrity, compliance, and stakeholder management. Traceability allows participants, business stakeholders, or consumers to manage and respond in a responsive and documented way. Even with the help of blockchains, achieving traceability among multiple blockchains is not an easy task. A dispute may arise due to ambiguities in contract clauses or the lack of accountability. All these matters should be based on an interoperable blockchain ecosystem that can mutually communicate with each other. While smart contracts can streamline processes within an organization, achieving interoperability between smart contracts from multiple organizations can be challenging.

Healthcare. One of the areas in which blockchain has tremendous influence is healthcare. Blockchain technology has great potential to transform the healthcare ecosystem to a new level. For example, interoperability can enhance clinical care service by offering access to historical clinical data even from other hospitals [31]. The landscape of health interoperability is primarily focused on special organizations such as hospitals and clinics, and internal information infrastructure usually creates sales details. However, many interoperability issues remain. For example, an interchange between separate organizations can be operationally difficult and requires substantial cooperation among different entities. Data-sharing agreements and procedures for patient matching should be agreed upon before actual data can be exchanged. Besides, there are numerous technical barriers, e.g., authentication and privacy-preserving schemes [51]. The aforementioned issues can be mitigated with interoperable blockchain systems.

Industry Process. Industrial processes typically require multiple entity collaborations, and each entity can build its blockchain system. Without information sharing, each blockchain system performs as an isolated island, where the potential collaboration is greatly limited. Blockchain

interoperability is highly required to create an interoperable platform with the guaranteed features of blockchain. However, when integrating interoperable blockchain systems into industrial use cases, it first needs to overcome several challenges, such as platform security, data privacy, and application-specificity. Current industrial blockchains lack a design standard and a collaborative environment to interoperate blockchains.

6.2 Decentralized Blockchain Internet

Following the design principles of the Internet, blockchain interoperability aims to create a decentralized Internet, in which each blockchain application can facilitate the packet switch communication without considering the underlying infrastructure. As pointed out in Reference [54], the Internet should have stable survivability, a variety of service types, and a variety of networks. Survivability ensures that connectivity across the Internet should not be compromised even under network failures. Services mean that the Internet must support multiple communications services. And, networks indicate that the Internet must accommodate a variety of networks on different scales.

Establishing decentralized Internet can distribute the power of authorities to users, which provides the fairness of participation or distribution of network resources. The construction requires different types of components in the perspective of blockchain's promise of decentralization, relating to basic components of decentralized naming services, discovery systems, routing schemes, and file storage [3, 4]. However, several challenges remain. For instance, designing a secure and distributed naming service, such as key management [9], is difficult due to the high cost and asynchronous updates. The routing mechanism should have to take care of various blockchain features and have the ability to route a transaction between different blockchain networks. One of the major concerns, for inter-blockchain network routing, is of verification of blockchain records and the provision of communication between any two peers belonging to two distinct blockchains. Decentralized storage requires that users can securely and privately store their data without disclosing it to any untrusted entities. There exist several decentralized storage solutions, e.g., Storj [132], **Inter-Planetary File System (IPFS)** [14]. The challenge of applying such situations to a large-scale decentralized Internet is known as scalability. Furthermore, a decentralized blockchain Internet infrastructure should have the feature of fault tolerance, e.g., survivability under blockchain failures. And, as each blockchain is a participant in a decentralized network, the blockchains must keep complexity and logic outside of the decentralized Internet.

6.3 Standardization

There are currently no standards for establishing compatible architectures for blockchain interoperability. Without an available standardization to regulate distinct blockchains, it is difficult to achieve a shared service principle. Moreover, every organization may develop incompatible standards against partners. This further blocks the progress of interoperability.

To achieve standardization, there are two possible directions: one is to formally agree to some practices that already have wide adoption, the so-called industry or de facto standards; while the other is to create a platform (e.g., by international standards development organizations) to allow competing interests to interoperate in various jurisdictions. As these implementations may have similar or overlapping functionalities by different organizations (or vendors), it is highly recommended to have international organizations regulate these processes.

Besides, interoperability is only in its early stages, and many research efforts need to be further explored. There is no doubt that a single party cannot have abilities to resolve all the issues of blockchain interoperability and coordinate the attempts of industry organizations and academic researchers. Although many promising examples of interoperability across multiple blockchain

systems are being developed, it is good to see these solutions are carried out based on decentralized databases instead of centralized ones. A uniform standard will greatly attract more developers to join this game and give effective contributions under the same principles.

7 CHALLENGES

This section explores challenges when achieving an interoperable blockchain ecosystem. Instead of discussing technical details, we pose them from a high-level perspective.

7.1 Survivability

Survivability is the key to the success of blockchain interoperability. It means that the transactions from an end-user application will be confirmed on a single ledger system or multiple ledger systems. The packets routing (in the form of transactions) through multiple domains must remain to be opaque to the targeted application and should be within a reasonable delay. In blockchain cases, the features of reliability and "best-effort delivery" are challenging to maintain, and it is hard to guarantee the application-level transaction can be completed within a reasonable time. Communications over multiple blockchain networks should be "connectionless," which means, from a high-level perspective, one blockchain does not need to consider if that transaction has been executed or not in another blockchain. However, one of the obstacles of application-level survivability is that a transaction may get executed and confirmed on a small portion of blockchains, however, this transaction should be confirmed independently on all related blockchains, and the application should be kept transparent for this process. This may cause an inconsistent state among multiple chains. Thus, an underlying mechanism to guarantee consistency among blockchains is necessary. These mechanisms should be transparent to its user "blockchain."

To achieve survivability, it also needs to handle many issues, e.g., reliability, semantic types, and distinguishability. The reliability of interoperable blockchains depends on where the function of reliability should be placed. For example, the retransmission mechanism (if transmission failed) can be enforced in different layers (such as the application layer, blockchain network layer, or even hidden middle layer). A semantic type means that different blockchain systems may have different ledger-level transactions, and these transactions may not be compatible with each other. Distinguishability of blockchain systems means that an application should be distinguishable from a group of interoperable blockchain systems, even if they all are semantics-compatible. Besides survivability, interoperable blockchain systems are supposed to support a variety of service types, such as processing speed, confirmation threshold, transaction directionality, and consensus finality.

7.2 Trustless Technology

A blockchain system itself can guarantee trustworthiness among participants via consensus protocols to ensure stable operations of the entire system. Even though no centralized authority has been involved, participants still believe that the network will operate as expected. But in multiple blockchain systems for interoperability, there is no such guarantee. In reality, multiple blockchain systems work independently, and to successfully proceed with communications, an intermediary "trusted" entity is required among distinct systems.

Many existing interoperability schemes remove the use of a single trusted entity and, instead, use a distributed trust. Trust is disseminated to a decentralized network, and no entity has the sole power of monopoly over the act of transacting. However, the absence of a single trusted entity in charge of coordinating interactions over networks does not, in and of itself, make it a "trustless technology." This trend of effort is still considered to be a centralized method. In such designs, the trust is still not completely removed but is shifted from a single intermediary to a federation

or committee. Essentially, a consensus over multiple blockchains is still required. Achieving fully decentralized communication, and thus fully eliminating the trust, among multiple blockchain systems, is still a challenging task.

7.3 From Theory to Practice

As discussed earlier, many blockchain application scenarios require interoperability with use cases from finance to industry and to economics. Most interoperability solutions are still in theory, or with prototype demonstration, and few have real implementations. One reason is that the theoretical advances on blockchain interoperability have not been agreed upon. Every organization may develop interoperable blockchain solutions based on its own requirements. This creates an isolated island and thus limits the achievement of theoretical efforts. Another reason is the absence of a global clock across multiple chains, which explicitly requires either agreement and trust of a third party, or reliance on a chain-dependent time definition, such as the block generation rate [49]. A practical implementation needs to consider different evaluation metrics such as throughput, latency, scalability, cost, security, and privacy that would help in speeding up the development process and overall advancement regarding interoperability. Many variants, such as consensus algorithm, computation and communication capabilities, or even peer-to-peer network delay, may affect a correct cross-chain operation. Especially if the timelock is used, assets may be locked forever. A real implementation depends on many timing-related factors. For example, protocols employing cross-chain verification rely on the timely arrival of proof and metadata, while in practice, it is hard to guarantee these timing factors. The lack of standards for related aspects affects the progress of blockchain interoperability. Thus, developers should conduct empirical studies and establish benchmarking data about the platform being developed.

7.4 Attacks Mitigation Technology

Typically, each independent blockchain has a well-defined security model of its own. A security model that works well in one blockchain may not be suitable in another. For example, blockchain X may rely on PoW and assume the adversarial hash computation is less than 50%, while another blockchain Y may adopt a PoS as its consensus protocol and assume that the stake of adversaries is less than 33%. Due to different criteria to evaluate the ability of an adversary, accumulated states may be lower than that of the accumulation computational power or vice versa [20]. Considering the permissionless public blockchains that are not Sybil resistant [43], achieving interoperability among blockchains will be difficult [140]. In a cross-chain setting, it is almost impossible to detect bribing attacks executed cross-chain [82].

Every blockchain should prevent replay attacks, where a transaction is submitted multiple times or on multiple chains. Replay attacks can result in failures, such as double-spending. In general, it is not hard to detect a replay attack in one blockchain. However, when involving multiple interoperable blockchains, the detection is difficult [111]. For example, in a single blockchain, protection may involve the use of a sequence number or may keep track of previously processed proofs. However, no similar mechanism exists in multiple blockchains [81]. Besides, multiple blockchain systems must carefully defend composability attacks, which are related to the consensus stability (which means the probability of a chain reversion is negligible [123]).

Besides, there exist many other challenges, especially for the compatibility of cryptographic primitives and collateralization. Different blockchains may leverage different cryptographic schemes, and compatible cryptographic primitives are highly required. However, it is hard to formalize the usage of compatible cryptographic primitives. Collateralization often occurs in cryptocurrency-related chains, which use a valuable asset (e.g., fiat money) to serve as an escrow [141]. It is crucial to guarantee that the available collateral has sufficient value to outweigh

potential gains from misbehaviors [140]. Moreover, blockchain interoperability faces standardization challenges. A well-authenticated and certified standard would require collaborations from international standardization authorities, and individual explorations may limit its target applications. Both theoretical and practical efforts are needed to enhance the standardization of interoperability.

8 POTENTIAL RESEARCH

In this section, we point out the potential research directions that may improve the availability of interoperation blockchain systems.

8.1 Interoperable Architecture

Given that many interoperability solutions are separated for specific purposes, no standards have been proposed. Until now, there is no formal definition of blockchain interoperability, nor an interoperable architecture. To promote the development of a decentralized Internet, it is necessary to first model blockchain communication at all involved layers (e.g., following the Internet's OSI model [136]). A modeling process must consider a variety of applications, since different applications may have distinct requirements. For example, when we model a transport layer protocol in TCP/IP stack, we may need to consider the choice of TCP or UDP. Thus, application-specific scenarios should be considered without affecting the overall interoperable blockchain architecture.

Due to the heterogeneity and diversity of services, it is a challenge to support complex data and processes. For example, applications may have different factors that affect overall interoperability, such as differences in consensus protocols, block sizes, and block generation. To establish an interoperable blockchain architecture, the designers, at least, need to consider several aspects, including distribution, big data, heterogeneity, dynamicity, and mobility. Furthermore, since the participants over a decentralized Internet would be geographically distributed, the data to be processed would be increased exponentially with an increasing number of participants. The systems of participants may be heterogeneous, and the services a participant accesses may be dynamic. All these factors affect the achievement of an interoperable architecture.

8.2 Cross-chain Primitives

Cross-chain interaction remains an active research direction. Our categorization, namely, *chain-based*, *bridge-based DApp-based* operability, can be considered as an independent model that satisfies partial requirements. Still, there may exist other cross-chain primitives for exploration. For example, we may consider combining the chain-based and bridge-based solutions to design a cross-chain communication using decentralized bridges, instead of using a two-way peg solution. Also, hybrid solutions may help to overcome drawbacks in specific solutions. For example, by integrating the sharding technology of a database system, a certain level of scalability and interoperability can be achieved among the sharded chains. However, blockchain sharding also requires the deployment of its cross-chain communication scheme [126].

The success of blockchain interoperabilities highly depends on the process of cross-chain communication and hence the related primitives. A good cross-chain primitive should have several features, such as eliminating a centralized entity and enabling multi-party transactions, reasonable performance, and a portable interface. Further, cross-chain primitives should not work against the decentralization principle of blockchains, as the centralized entity may become a performance bottleneck and an attack target, e.g., a single point of failure. Most existing cross-chain schemes still focus on two-party scenarios, whereas a multi-party communication model is still missing. Considering the performance of cross-chain communications, an acceptable response time will also be required. The cross-chain primitives are portable. For example, when participants concurrently

work with multiple cross-chain platforms, portability issues occur from different interfaces. Thus, a common interface with different blockchains and their participants is preferable. Besides, a good cross-chain primitive should be provable, secure, correct, and atomic. Also, it requires strict proof to theoretically prove it works.

8.3 Security and Privacy

When multiple blockchains work together, security and privacy are necessary considerations. Different blockchains may adopt different security primitives, in which one security primitive is secure in one system, while not secure in another system. It is highly recommended to develop security standards for scripting smart contracts and other blockchain primitives, as security is still the major concern for the willingness to adopt interoperability by stakeholders. Meanwhile, privacy is crucial in any sensitive interaction (e.g., financial assets or health records) and thus in cross-chain communication. The privacy-preserving technologies in current blockchain systems are not sufficiently robust. The ideal solution for preserving privacy in multiple blockchain systems would be in a form of confidentially decentralized record-keeping, which is completely obfuscated and anonymous by design.

In practice, different applications have different security primitives to guarantee secure operations, and it is hard, if not impossible, to make all applications adopt the same security primitive. When integrating them together, new data from an arbitrary process may go far beyond the outreach of any common security safeguard. This may make the data and services vulnerable. With the great adoption of emerging mobile devices, there is a huge concern for secure information transfer and message exchange between different blockchain systems. For example, insecure blockchain systems may initiate a cross-chain transaction to a secure blockchain system, to acquire sensitive data, or even launch an attack on secure blockchain systems. Thus, security and privacy-assisted technologies are required to succeed in blockchain interoperability.

8.4 Scalability

Scalability is not only an issue of blockchain interoperability but also an issue of blockchain itself. The key features (e.g., decentralization and immutability) of a blockchain require that every full node store a full copy of the blockchain; however, this comes at a cost of scalability. The scalability issue limits the wide usage of blockchain in large-scale networks. Typically, scalability can be evaluated by the *throughput* (e.g., measured by transactions per second) against the number of participating nodes and the size of concurrent workloads [41, 42].

In current designs, many blockchain systems are still suffering from poor throughput. Scaling blockchain has become an active research area [35]. Blockchain interoperability and scalability have close relations with the design of blockchain architecture, and they can share the same design structure. In general, there are several methods to scale blockchain, e.g., on-chain, off-chain, side-chain, child-chain, and inter-chain solutions [65]. An on-chain solution modifies only elements within a blockchain to increase scalability. An off-chain solution processes the transactions outside of the blockchain, e.g., as a state-channel solution, by maintaining the state of the main chain. The side-chain solution exchanges assets of different blockchains with each other; its structure is similar to the description in Section 4.1. The child-chain solution has a parent-child structure, processes transactions in the child-chain, and records results in the parent-chain, as the structure of SMChain in Section 4.3. The Inter-chain solution provides a way to enable communication among the various blockchains, whose infrastructure is like the side-chain solution.

Other techniques are also promising, such as via increased block size [50], using DAG techniques [130] and sharding techniques [126]. Besides the above-mentioned research directions on

blockchain interoperability, there are other hot research areas, e.g., standardization, usability, and reachability.

9 CONCLUSION

The development of blockchain interoperability is still in the infancy stage. This article presents a Systematization of Knowledge for existing efforts in such a field. We classify them into several key categories, namely, chain-based interoperability, bridge-based interoperability, and DApp-based interoperability. For each category, we have studied state-of-the-art solutions with detailed analysis in terms of both advantages and disadvantages. This article serves as a starting point for exploring blockchain interoperability. Based on our observation, we discuss opportunities and challenges when applying blockchain interoperability to the current blockchain design. Finally, we provide research directions that may help to advance an interoperable blockchain ecosystem.

ACKNOWLEDGMENT

A primary version has been released at https://eprint.iacr.org/2021/537.

REFERENCES

[1] Tron Team. 2022. Tron network. Retrieved from https://tron.network/.
[2] Alfarez Abdul-Rahman and Stephen Hailes. 1998. A distributed trust model. In *Workshop on New Security Paradigms*. 48–60.
[3] Anwaar Ali, Siddique Latif, Junaid Qadir, Salil Kanhere, Jatinder Singh, Jon Crowcroft, et al. 2019. Blockchain and the future of the internet: A comprehensive review. *arXiv preprint arXiv:1904.00733* (2019).
[4] Muneeb Ali, Ryan Shea, Jude Nelson, and Michael J. Freedman. 2017. Blockstack: A new decentralized internet. *Whitepaper* (2017). https://pdos.csail.mit.edu/6.824/papers/blockstack-2017.pdf.
[5] Muhammad Salek Ali, Massimo Vecchio, Miguel Pincheira, Koustabh Dolui, Fabio Antonelli, and Mubashir Husain Rehmani. 2018. Applications of blockchains in the Internet of Things: A comprehensive survey. *IEEE Commun. Surv. Tutor.* 21, 2 (2018), 1676–1717.
[6] Mohammad Hassan Ameri, Mahshid Delavar, Javad Mohajeri, and Mahmoud Salmasizadeh. 2018. A key-policy attribute-based temporary keyword search scheme for secure cloud storage. *IEEE Trans. Cloud Comput.* 8, 3 (2018), 660–671.
[7] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. CAPER: A cross-application permissioned blockchain. *Proc. VLDB Endow.* 12, 11 (2019), 1385–1398.
[8] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich et al. 2018. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *13th EuroSys Conference*.ACM.
[9] Stefano Angieri, Alberto García-Martínez, Bingyang Liu, Zhiwei Yan, Chuang Wang, and Marcelo Bagnulo. 2019. A distributed autonomous organization for Internet address management. *IEEE Trans. Eng. Manag.* 67, 4 (2019), 1459–1475.
[10] ARK. 2019. ARK ecosystem whitepaper. *Version 2.1.0* Retrieved from https://ark.io/Whitepaper.pdf.
[11] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. 2014. Enabling blockchain innovations with pegged sidechains. Retrieved from http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains.
[12] Deme Balazs. 2017. Herdius whitepaper. Retrieved from https://herdius.com/whitepaper/Herdius_Technical_Paper.pdf.
[13] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A survey on blockchain interoperability: Past, present, and future trends. *ACM Comput. Surv.* 54, 8 (2021), 1–41.
[14] Juan Benet. 2014. IPFS-content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561* (2014).
[15] Iddo Bentov, Yan Ji, Fan Zhang, Lorenz Breidenbach, Philip Daian, and Ari Juels. 2019. Tesseract: Real-time cryptocurrency exchange using trusted hardware. In *ACM SIGSAC Conference on Computer and Communications Security*. 1521–1538.
[16] Rajesh Bhatia et al. 2020. Interoperability solutions for blockchain. In *International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*. IEEE, 381–385.
[17] Binance. 2019. Binance chain (DEX). *Version 1.1*. Retrieved from https://docs.binance.org/.

[18] Matthew Black, TingWei Liu, and Tony Cai. 2019. Atomic loans: Cryptocurrency debt instruments. *arXiv preprint arXiv:1901.05117* (2019).
[19] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. 2018. Verifiable delay functions. In *Annual International Cryptology Conference*. Springer, 757–788.
[20] Joseph Bonneau, Edward W. Felten, Steven Goldfeder, Joshua A. Kroll, and Arvind Narayanan. 2016. Why buy when you can rent? Bribery attacks on Bitcoin consensus. In *Financial Cryptography and Data Security (FC'16)*. Springer, 19–26.
[21] Michael Borkowski, Philipp Frauenthaler, Marten Sigwart, Taneli Hukkinen, Oskar Hladky, and Stefan Schulte. 2019. Cross-blockchain technologies: Review, state-of-the-art, and outlook. *White Paper* (2019). https://dsg.tuwien.ac.at/tast/pub/tast-white-paper-4.pdf.
[22] Michael Borkowski, Daniel McDonald, Christoph Ritzer, and Stefan Schulte. 2018. *Towards Atomic Cross-chain Token Transfers: State of the Art and Open Questions within TAST.* Distributed Systems Group TU Wien (Technische Universit at Wien), Report.
[23] Michael Borkowski, Christoph Ritzer, Daniel McDonald, and Stefan Schulte. 2018. Caught in chains: Claim-first transactions for cross-blockchain asset transfers. *Technische Universität Wien, Whitepaper* (2018). https://dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-2.pdf.
[24] Michael Borkowski, Marten Sigwart, Philipp Frauenthaler, Taneli Hukkinen, and Stefan Schulte. 2019. DeXTT: Deterministic cross-blockchain token transfers. *IEEE Access* 7 (2019), 111030–111042.
[25] Danny Bradbury. 2013. The problem with Bitcoin. *Comput. Fraud Secur.* 2013, 11 (2013), 5–8.
[26] Ethan Buchman. 2016. *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains.* Ph.D. Dissertation. University of Guelph.
[27] Jeff Burdges, Alfonso Cevallos, Peter Czaban, Rob Habermeier, Syed Hosseini, Fabio Lama, Handan Kilinc Alper, Ximin Luo, Fatemeh Shirazi, Alistair Stewart, et al. 2020. Overview of Polkadot and its design considerations. *arXiv preprint arXiv:2005.13456* (2020).
[28] Vitalik Buterin. 2016. Chain interoperability. *R3 Research Paper* (2016). https://theblockchaintest.com/uploads/resources/R3%20-%20Chain%20Interoperability%20-%202016%20-%20Sep.pdf.
[29] Christian Cachin and Marko Vukolić. 2017. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873* (2017).
[30] Miguel Castro, Barbara Liskov, et al. 1999. Practical Byzantine fault tolerance. In *OSDI*, USENIX Association, Vol. 99. 173–186.
[31] Bipartisan Policy Center. 2012. Clinician perspectives on electronic health information sharing for transitions of care. Bipartisan Policy Center, Washington, DC. https://bipartisanpolicy.org/download/?file=/wp-content/uploads/2019/03/Clinician-Survery_format-2.pdf.
[32] Joseph Chow. 2016. Btc relay. *btc-relay* (2016). https://archive.devcon.org/archive/watch/1/btc-relay/?tab=YouTube.
[33] COMIT. 2020. COMIT protocol stack. Retrieved from https://comit.network/docs/comit-protocol/comit-protocol-stack/.
[34] Flaviu Cristian. 1996. Synchronous and asynchronous. *Commun. ACM* 39, 4 (1996), 88–97.
[35] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. 2016. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*. Springer, 106–125.
[36] Arlyn Culwick and Dan Metcalf. 2018. The Blocknet design specification. https://blocknet.co/whitepaper/Blocknet_Whitepaper.pdf.
[37] Bingrong Dai, Shengming Jiang, Menglu Zhu, Ming Lu, Dunwei Li, and Chao Li. 2020. Research and implementation of cross-chain transaction model based on improved hash-locking. In *International Conference on Blockchain and Trustworthy Systems*. Springer, 218–230.
[38] Liping Deng, Huan Chen, Jing Zeng, and Liang-Jie Zhang. 2018. Research on cross-chain technology based on sidechain and hash-locking. In *International Conference on Edge Computing*. Springer, 144–151.
[39] Johnny Dilley, Andrew Poelstra, Jonathan Wilkins, Marta Piekarska, Ben Gorlick, and Mark Friedenbach. 2016. Strong federations: An interoperable blockchain solution to centralized third-party risks. *arXiv preprint arXiv:1612.05491* (2016).
[40] Sheng Ding, Jin Cao, Chen Li, Kai Fan, and Hui Li. 2019. A novel attribute-based access control scheme using blockchain for IoT. *IEEE Access* 7 (2019), 38431–38441.
[41] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. 2018. Untangling blockchain: A data processing view of blockchain systems. *IEEE Trans. Knowl. Data Eng.* 30, 7 (2018), 1366–1385.
[42] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. Blockbench: A framework for analyzing private blockchains. In *ACM International Conference on Management of Data*. ACM, 1085–1100.

[43] Tuyet Duong, Lei Fan, and Hong-Sheng Zhou. 2016. 2-hop blockchain: Combining proof-of-work and proof-of-stake securely. *Cryptology ePrint Archive, Report 2016/716* (2016).
[44] Christoph Egger, Pedro Moreno-Sanchez, and Matteo Maffei. 2019. Atomic multi-channel updates with constant collateral in bitcoin-compatible payment-channel networks. In *ACM SIGSAC Conference on Computer and Communications Security*. 801–815.
[45] Ghareeb Falazi, Uwe Breitenbücher, Florian Daniel, Andrea Lamparelli, Frank Leymann, and Vladimir Yussupov. 2020. Smart contract invocation protocol (SCIP): A protocol for the uniform integration of heterogeneous blockchain smart contracts. In *International Conference on Advanced Information Systems Engineering*. Springer, 134–149.
[46] Philipp Frauenthaler, Michael Borkowski, and Stefan Schulte. 2020. A framework for assessing and selecting blockchains at runtime. In *IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. IEEE, 106–113.
[47] Philipp Frauenthaler, Marten Sigwart, Christof Spanring, and Stefan Schulte. 2020. Testimonium: A cost-efficient blockchain relay. *arXiv preprint arXiv:2002.12837* (2020).
[48] Enrique Fynn, Alysson Bessani, and Fernando Pedone. 2020. Smart contracts on the move. *arXiv preprint arXiv:2004.05933* (2020).
[49] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2017. The bitcoin backbone protocol with chains of variable difficulty. In *Annual International Cryptology Conference*. Springer, 291–323.
[50] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the security and performance of proof of work blockchains. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 3–16.
[51] William J. Gordon and Christian Catalini. 2018. Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Computat. Struct. Biotechnol. J.* 16 (2018), 224–230.
[52] Runchao Han, Haoyu Lin, and Jiangshan Yu. 2019. On the optionality and fairness of atomic swaps. In *1st ACM Conference on Advances in Financial Technologies*. 62–75.
[53] Theo Härder. 2005. DBMS architecture–still an open problem. *Datenbanksysteme in Business, Technologie und Web, 11. Fachtagung des GIFachbereichs "Datenbanken und Informationssysteme" (DBIS)*. https://dl.gi.de/handle/20.500.12116/28295.
[54] Thomas Hardjono, Alexander Lipton, and Alex Pentland. 2018. Towards a design philosophy for interoperable blockchain systems. *arXiv preprint arXiv:1805.05934* (2018).
[55] Thomas Hardjono, Alexander Lipton, and Alex Pentland. 2019. Toward an interoperability architecture for blockchain autonomous systems. *IEEE Trans. Eng. Manag.* (2019).
[56] Maurice Herlihy. 2018. Atomic cross-chain swaps. In *ACM Symposium on Principles of Distributed Computing*. 245–254.
[57] Maurice Herlihy, Barbara Liskov, and Liuba Shrira. 2019. Cross-chain deals and adversarial commerce. *arXiv preprint arXiv:1905.09743* (2019).
[58] Hyperledger. 2020. Hyperledger cactus whitepaper. Retrieved from https://github.com/hyperledger/cactus.
[59] Interledger. 2020. Interledger protocol V4. (2020). Retrieved from https://interledger.org/rfcs/0027-interledger-protocol-4/.
[60] Hai Jin, Xiaohai Dai, and Jiang Xiao. 2018. Towards a novel architecture for enabling interoperability amongst multiple blockchains. In *IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1203–1211.
[61] Luo Kan, Yu Wei, Amjad Hafiz Muhammad, Wang Siyuan, Gao Linchao, and Hu Kai. 2018. A multiple blockchains architecture on inter-blockchain communication. In *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 139–145.
[62] Majid Khabbazian, Tejaswi Nadahalli, and Roger Wattenhofer. 2019. Outpost: A responsive lightweight watchtower. In *1st ACM Conference on Advances in Financial Technologies*. 31–40.
[63] Minhaj Ahmad Khan and Khaled Salah. 2018. IoT security: Review, blockchain solutions, and open challenges. *Fut. Gener. Comput. Syst.* 82 (2018), 395–411.
[64] Aggelos Kiayias, Nikolaos Lamprou, and Aikaterini-Panagiota Stouka. 2016. Proofs of proofs of work with sublinear complexity. In *International Conference on Financial Cryptography and Data Security*. Springer, 61–78.
[65] Soohyeong Kim, Yongseok Kwon, and Sunghyun Cho. 2018. A survey of scalability solutions on blockchain. In *International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 1204–1207.
[66] Matthias Kühne. 2020. *Extending Cross-blockchain Token Transfers*. Ph.D. Dissertation. Wien.
[67] Jae Kwon and Ethan Buchman. 2016. Cosmos: A network of distributed ledgers. Retrieved from https://cosmos.network/whitepaper.
[68] Jae Kwon and Ethan Buchman. 2019. Cosmos whitepaper. https://cosmos.network/whitepaper.
[69] Pascal Lafourcade and Marius Lombard-Platet. 2020. About blockchain interoperability. *Inform. Process. Lett.* 161 (2020), 105976.

[70] Daniel Larimer. 2014. Delegated proof-of-stake (DPOS). *Bitshare Whitepaper* (2014). https://blog.bitmex.com/wp-content/uploads/2018/06/173481633-BitShares-White-Paper.pdf.
[71] Sergio Demian Lerner. 2015. RSK White paper overview. (2015). https://rootstock.io/rsk-white-paper-updated.pdf.
[72] Sergio Damian Lerner. 2016. Drivechains, sidechains and hybrid 2-way peg designs. (2016). https://docs.rsk.co/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf.
[73] Zhetao Li, Jiawen Kang, Rong Yu, Dongdong Ye, Qingyong Deng, and Yan Zhang. 2017. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Trans. Industr. Inform.* 14, 8 (2017), 3690–3700.
[74] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. 2017. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* 4, 5 (2017), 1125–1142.
[75] Zhuotao Liu, Yangxi Xiang, Jian Shi, Peng Gao, Haoyu Wang, Xusheng Xiao, Bihan Wen, and Yih-Chun Hu. 2019. Hyperservice: Interoperability and programmability across heterogeneous blockchains. In *ACM SIGSAC Conference on Computer and Communications Security*. 549–566.
[76] Ankur Lohachab, Saurabh Garg, Byeong Kang, Muhammad Bilal Amin, Junmin Lee, Shiping Chen, and Xiwei Xu. 2021. Towards interconnected blockchains: A comprehensive review of the role of interoperability among disparate blockchains. *ACM Comput. Surv.* 54, 7 (2021), 1–39.
[77] Loom. 2016. Intro to Loom Network. Loom SDK. Retrieved from https://loomx.io/developers/en/intro-to-loom.html.
[78] Loi Luu, Nate Rush, and Nicholas Lin. 2019. PeaceRelay: Connecting the many Ethereum blockchains. *Retrieved Oct* 15 (2019), 2019. https://github.com/KyberNetwork/peace-relay.
[79] Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. 2017. Concurrency and privacy with payment-channel networks. In *ACM SIGSAC Conference on Computer and Communications Security*. 455–471.
[80] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. 2019. Anonymous multi-hop locks for blockchain scalability and interoperability. In *Network and Distributed System Security Symposium*.
[81] Patrick McCorry, Ethan Heilman, and Andrew Miller. 2017. Atomically trading with Roger: Gambling on the success of a hardfork. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 334–353.
[82] Patrick McCorry, Alexander Hicks, and Sarah Meiklejohn. 2018. Smart contracts for bribing miners. In *International Conference on Financial Cryptography and Data Security*. Springer, 3–18.
[83] Andrew Miller, Iddo Bentov, Ranjit Kumaresan, and Patrick McCorry. 2017. Sprites: Payment channels that go faster than lightning. *CoRR abs/1702.05812* 306 (2017).
[84] Mahdi H. Miraz and David C. Donald. 2019. Atomic cross-chain swaps: Development, trajectory and potential of non-monetary digital token swap facilities. *Ann. Emerg. Technol. Comput.* 3 (2019).
[85] Satoshi Nakamoto et al. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008). https://bitcoin.org/bitcoin.pdf.
[86] POA Network. 2018. POA-network-whitepaper. *Accessed* 10, 3 (2018), 19. https://www.poa.network/v/master-1/.
[87] Jonas Nick, Andrew Poelstra, and Gregory Sanders. 2019. Liquid: A strongly federated asset issuance platform. (2019). https://blog.blockstream.com/en-strong-federations-paper-released-liquid/.
[88] William Nikolakis, Lijo John, and Harish Krishnan. 2018. How blockchain can shape sustainable global value chains: An evidence, verifiability, and enforceability (EVE) framework. *Sustainability* 10, 11 (2018), 3926.
[89] Markus Nissl, Emanuel Sallinger, Stefan Schulte, and Michael Borkowski. 2020. Towards cross-blockchain smart contracts. *arXiv preprint arXiv:2010.07352* (2020).
[90] Tier Nolan. 2013. Alt chains and atomic transfers. In *Bitcoin Forum*. https://bitcointalk.org/index.php?topic=193281.0.
[91] Babu Pillai, Kamanashis Biswas, and Vallipuram Muthukkumarasamy. 2019. Blockchain interoperable digital objects. In *International Conference on Blockchain*. Springer, 80–94.
[92] Joseph Poon and Vitalik Buterin. 2017. Plasma: Scalable autonomous smart contracts. *White Paper* (2017), 1–47. https://www.plasma.io/plasma.pdf.
[93] Joseph Poon and Thaddeus Dryja. 2016. The bitcoin lightning network: Scalable off-chain instant payments. (2016). https://lightning.network/lightning-network-paper.pdf.
[94] Dan Pritchett. 2008. Base: An acid alternative. *Queue* 6, 3 (2008), 48–55.
[95] Aleksei Pupyshev, Elshan Dzhafarov, Ilya Sapranidi, Inal Kardanov, Shamil Khalilov, and Sten Laureyssens. 2020. SuSy: A blockchain-agnostic cross-chain asset transfer gateway protocol based on Gravity. *arXiv preprint arXiv:2008.13515* (2020).
[96] Aleksei Pupyshev, Dmitry Gubanov, Elshan Dzhafarov, Inal Kardanov, Vladimir Zhuravlev, Shamil Khalilov, Marc Jansen, Sten Laureyssens, Igor Pavlov, Sasha Ivanov, et al. 2020. Gravity: A blockchain-agnostic cross-chain communication and data oracles protocol. *arXiv preprint arXiv:2007.00966* (2020).
[97] Ilham A. Qasse, Manar Abu Talib, and Qassim Nasir. 2019. Inter blockchain communication: A survey. In *ArabWIC 6th Annual International Conference Research Track*. 1–6.

[98] Kaihua Qin and Arthur Gervais. 2018. An overview of blockchain scalability, interoperability and sustainability. *Hochschule Luzern Imperial College London Liquidity Network* (2018). https://www.eublockchainforum.eu/sites/default/files/research-paper/an_overview_of_blockchain_scalability_interoperability_and_sustainability.pdf.
[99] Ronald L. Rivest, Adi Shamir, and David A. Wagner. 1996. Time-lock puzzles and timed-release crypto. (1996). https://people.csail.mit.edu/rivest/pubs/RSW96.pdf.
[100] Peter Robinson. 2020. Consensus for crosschain communications. *arXiv preprint arXiv:2004.09494* (2020).
[101] Eder Scheid, Bruno Rodrigues, and Burkhard Stiller. 2019. Toward a policy-based blockchain agnostic framework. In *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 609–613.
[102] Eder J. Scheid, Timo Hegnauer, Bruno Rodrigues, and Burkhard Stiller. 2019. Bifröst: A modular blockchain interoperability API. In *IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 332–339.
[103] Eder J. Scheid, Daniel Lakic, Bruno B. Rodrigues, and Burkhard Stiller. 2020. PleBeuS: A policy-based blockchain selection framework. In *IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–8.
[104] Stefan Schulte, Marten Sigwart, Philipp Frauenthaler, and Michael Borkowski. 2019. Towards blockchain interoperability. In *International Conference on Business Process Management*. Springer, 3–10.
[105] Ori Shalev and Nir Shavit. 2006. Split-ordered lists: Lock-free extensible hash tables. *J ACM* 53, 3 (2006), 379–405.
[106] Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon. 2010. *Handbook of Peer-to-peer Networking*. Vol. 34. Springer Science & Business Media.
[107] Marten Sigwart, Philipp Frauenthaler, Christof Spanring, and Stefan Schulte. Towards cross-blockchain smart contracts. (n. d.). https://arxiv.org/pdf/2010.07352.pdf.
[108] Amritraj Singh, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *J. Netw. Comput. Applic.* 149 (2020), 102471.
[109] Vasilios A. Siris, Pekka Nikander, Spyros Voulgaris, Nikos Fotiou, Dmitrij Lagutin, and George C. Polyzos. 2019. Interledger approaches. *IEEE Access* 7 (2019), 89948–89966.
[110] Dale Skeen. 1981. Nonblocking commit protocols. In *ACM SIGMOD International Conference on Management of Data*. 133–142.
[111] Alberto Sonnino, Shehar Bano, Mustafa Al-Bassam, and George Danezis. 2020. Replay attacks and defenses against cross-shard consensus in sharded distributed ledgers. In *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 294–308.
[112] Matthew Spoke, NE Team, et al. 2017. Aion: Enabling the decentralized internet. *AION, White Paper, Jul* (2017). https://aion.theoan.com/.
[113] Michael Szydlo. 2004. Merkle tree traversal in log space and time. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 541–554.
[114] Stefan Tai, Jacob Eberhardt, and Markus Klems. 2017. Not acid, not base, but salt. In *7th International Conference on Cloud Computing and Services Science*. SCITEPRESS-Science and Technology Publications, Lda, 755–764.
[115] Block Collider Team. 2018. Block Collier whitepaper. *Version 0.9.9* Retrieved from https://overline.network/blockcollider_litepaper.pdf.
[116] Perun Team. 2020. Hyperledger labs. Retrieved from https://www.hyperledger.org/category/hyperledger-labs.
[117] Jason Teutsch, Michael Straka, and Dan Boneh. 2019. Retrofitting a two-way peg between blockchains. *arXiv preprint arXiv:1908.03999* (2019).
[118] Stefan Thomas and Evan Schwartz. 2015. A protocol for interledger payments. Retrieved from https://interledger.org/interledger.pdf.
[119] Itay Tsabary, Matan Yechieli, and Ittay Eyal. 2020. MAD-HTLC: Because HTLC is crazy-cheap to attack. *arXiv preprint arXiv:2006.12031* (2020).
[120] Ron van der Meyden. 2019. On the specification and verification of atomic swap smart contracts. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 176–179.
[121] Gilbert Verdian, Paolo Tasca, Colin Paterson, and Gaetano Mondelli. 2018. Quant overledger whitepaper. (2018). https://uploads-ssl.webflow.com/6006946fee85fda61f666256/60211c93f1cc59419c779c42_Quant_Overledger_Whitepaper_Sep_2019.pdf.
[122] Fabian Vogelsteller and Vitalik Buterin. 2015. Eip 20: Erc-20 token standard. (2015). https://eips.ethereum.org/EIPS/eip-20.
[123] Marko Vukolic. 2016. Eventually returning to strong consistency. *IEEE Data Eng. Bull.* 39, 1 (2016), 39–44.
[124] Wanchain. 2017. Wanchain: Building super financial markets for the new digital economy. *Whitepaper Version 0.9.1* Retrieved from https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf.
[125] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. 2019. SMChain: A scalable blockchain protocol for secure metering systems in distributed industrial plants. In *International Conference on Internet of Things Design and Implementation*. ACM, 249–254.

[126] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. 2019. SoK: Sharding on blockchain. In *1st ACM Conference on Advances in Financial Technologies*. ACM, 41–61.
[127] Hui Wang, Yuanyuan Cen, and Xuefeng Li. 2017. Blockchain router: A cross-chain communication protocol. In *6th International Conference on Informatics, Environment, Energy and Applications*. 94–97.
[128] Ke Wang, Zhizhe Zhang, and Hyong S. Kim. 2018. ReviewChain: Smart contract based review system with multi-blockchain gateway. In *IEEE International Conference on Internet of Things (iThings)*. IEEE, 1521–1526.
[129] Qin Wang, Rujia Li, Qi Wang, Shiping Chen, Mark Ryan, and Thomas Hardjono. 2022. Exploring web3 from the view of blockchain. *arXiv preprint arXiv:2206.08821* (2022).
[130] Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. 2020. SoK: Diving into DAG-based blockchain systems. *arXiv preprint arXiv:2012.06128* (2020).
[131] Peter Wegner. 1996. Interoperability. *ACM Comput. Surv.* 28, 1 (1996), 285–287.
[132] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, and Vitalik Buterin. 2014. Storj a peer-to-peer cloud storage network. (2014). https://www.storj.io/storj2014.pdf.
[133] Gavin Wood. 2016. Polkadot: Vision for a heterogeneous multi-chain framework. *White Paper* (2016). https://polkadot.network/whitepaper/.
[134] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ether. Proj. Yell. Pap.* 151, 2014 (2014), 1–32.
[135] Chao Xie, Chunzhi Su, Manos Kapritsos, Yang Wang, Navid Yaghmazadeh, Lorenzo Alvisi, and Prince Mahajan. 2014. Salt: Combining ACID and BASE in a distributed database. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*. 495–509.
[136] Yechiam Yemini. 1993. The OSI network management model. *IEEE Commun. Mag.* 31, 5 (1993), 20–29.
[137] Bin Yu, Shabnam Kasra Kermanshahi, Amin Sakzad, and Surya Nepal. 2019. Chameleon hash time-lock contract for privacy preserving payment channel networks. In *International Conference on Provable Security*. Springer, 303–318.
[138] Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. 2019. Atomic commitment across blockchains. *arXiv preprint arXiv:1905.02847* (2019).
[139] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. 2021. SoK: Communication across distributed ledgers. In *International Conference on Financial Cryptography and Data Security (FC)*. Springer, 3–36.
[140] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. 2021. SoK: Communication across distributed ledgers. In *Financial Cryptography and Data Security (FC'21)*. Springer, 3–36.
[141] Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, and William Knottenbelt. 2019. XCLAIM: Trustless, interoperable, cryptocurrency-backed assets. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 193–210.
[142] Yuhui Zhang, Dejun Yang, and Guoliang Xue. 2019. Cheapay: An optimal algorithm for fee minimization in blockchain-based payment channel networks. In *IEEE International Conference on Communications (ICC)*. IEEE, 1–6.
[143] Dongfang Zhao and Tonglin Li. 2020. Distributed cross-blockchain transactions. *arXiv preprint arXiv:2002.11771* (2020).
[144] Jean-Yves Zie, Jean-Christophe Deneuville, Jérémy Briffaut, and Benjamin Nguyen. 2019. Extending atomic cross-chain swaps. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 219–229.

Received 9 May 2022; revised 7 January 2023; accepted 30 January 2023