SoK: Exploring Blockchains Interoperability

Gang Wang
University of Connecticut
Email: email.gang.wang@gmail.com

***Abstract*—Distributed ledger technologies like blockchain have gained great attention in both academia and industry. Blockchain as a potentially disruptive technology can advance many different fields, e.g., cryptocurrencies, supply chains, and the industrial Internet of Things. The next-generation blockchain ecosystem is expected to consist of various homogeneous and heterogeneous distributed ledgers. These ledger systems will inevitably require a certain level of proper cooperation of multiple blockchains to enrich advanced functionalities and enhance interoperable capabilities for future applications. The interoperability among blockchains will definitely revolutionize current blockchain design principles, like the emergence of Internet. The development of cross-blockchain applications involves much complexity regarding the variety of underlying cross-blockchain communication. The way to effectively enable interoperability across multiple blockchains is thus essential and expecting to confront various unprecedented challenges. For instance, due to different transaction structures, ensuring the properties of ACID (Atomicity, Consistency, Isolation, Durability) in transactions processing and verification processes across diverse blockchain systems remains a challenging task in both academia and industry. This paper provides a systematic and comprehensive review of the current progress of blockchain interoperability. We explore both general principles and practical schemes to achieve interoperable blockchain systems. We then survey and compare the state-of-the-art solutions to deal with the interoperability of blockchains in detail. Finally, we discuss several critical challenges and some potential research directions to advance the research on exploring blockchain interoperability.**

I. Introduction

Blockchain has become a key enabler for implementing and advancing distributed ledgers. It allows a group of participating nodes (or parties) that do not trust each other to provide trustworthy and immutable services. Distributed ledgers were initially used as tamper-evident logs to record data. They are typically maintained by independent parties without a central authority. Blockchain became popular because of its success in crypto-currencies, e.g., Bitcoin [1]. Emerging blockchain technological advances and applications have earned tremendous attention from both industrial and academic domains, promising to change all aspects of digital business in the industry. Blockchain is a kind of Decentralized Ledger Technology (DLT) that heavily relies on cryptographic primitives to provide an immutable and verifiable data platform [2]. It is believed that blockchain will have a profound impact and influence on existing Internet infrastructures and promote the development of a decentralized Internet.

Obviously, different blockchain applications have different criteria, necessitating distinct blockchain capabilities and requirements. Due to the existence of various protocols and technologies, information cannot be exchanged freely and directly between two blockchains. The development of independent and incompatible blockchain technologies has caused significant fragmentation of the research, since users and developers have to choose from a set of blockchains for their own use case scenarios. This has led to incompatibility and isolation in today's blockchain ecosystem, and we see many distinct blockchains. It is desirable to achieve interoperable blockchain systems to freely exchange information in future infrastructures, e.g., as a global value-exchange network [3]. Originally, interoperability was described as the ability of two or more components to work together despite the existence of differences in language, interface, and execution environment [4]. In the context of blockchain, interoperability means connecting multiple blockchains to access information and act on it by changing its state or the state of another blockchain. Optimally, this would be achieved without compromising the blockchain's premise in decentralization and trustworthiness [5].

Blockchain interoperability would enable secure state information transitions across different blockchains, either homogeneous or heterogeneous, and create invaluable channels for connecting the decentralized Internet. Most existing proposals on blockchain interoperability focus on the process of atomic token exchange across blockchains (e.g., from a source blockchain system to a target blockchain system), with the goal of removing the need for centralized exchanges [6]. To achieve this atomic process, the token transferring must be in an autonomous and synchronized process among the involved blockchains without the help of a centralized entity. Practically speaking, there is no efficient way to fully replicate or duplicate the state of one blockchain to another blockchain [7]. To complete this process, some efficient schemes are required to perform the verification of information occurring on another blockchain, without the help from some trusted authorities [8]. An atomic swap is one technique that enables users of different blockchains to exchange their assets in an atomic and trustless manner [9]. One of the most popular scenarios is the atomic token swap. However, atomic token swapping protocols [10] are not self-inclusive enough to complete tasks of cross-chain decentralized applications (*dApps*) because the "executable" components in those dApps may involve more complex activities (e.g., verifying against historical information) than pure token transfers. For example, the atomic swapping process typically does not have the ability to destroy a certain amount of assets (e.g., in the form of tokens) in the source blockchain and re-create the same amount of transferred tokens on the target blockchain. In general, an atomic swap, as its name implies, offers only token exchanges rather than transfers. Also, this process always requires a counterparty (of another blockchain) who is willing to exchange these tokens [8].

Blockchain interoperability requires either that assets can be moved from one blockchain platform to another or that the users have the ability to access information from one blockchain inside another, without any additional efforts from a third party. Currently, the notion of blockchain interoperability is still in the conceptual stage and has had little practice, since successful blockchain interoperability requires at least two blockchains to freely exchange information, the way information is exchanged via the Internet. It not only needs to consider public blockchains, but also needs to cooperate with private and consortium blockchains. However, due to the security and privacy involved, private and consortium blockchains may not willing to share their information [11]. It would be highly desirable to provide a generic framework to cover most existing blockchain

systems. Thus, before achieving a successfully interoperable multi-blockchain system, many other challenges must be overcome (e.g., scalability when applying to a large-scale scenario [12]).

Interoperable blockchains would create a prototype of the decentralized Internet, and users of this Internet would be able to freely and directly exchange information with the guaranteed properties of the blockchain. Blockchains equipped with the feature of interoperability would improve the flexibility of interoperable ledger systems, and this will also provide a "bridge" to perform open assets exchanging without jeopardizing the smart function of blockchains. But, although blockchain interoperability is promising, it still faces many design challenges. A systematical-level study on blockchain interoperability is thus required. Generally speaking, there are some nice literature discussing chain interoperability in general. For example, Buterin [13] classified chain interoperability into three primary categories, namely, centralized or multisig notary schemes, sidechains/relays, and hash-locking; Belchior et al. [14] classified blockchain interoperability in three major categories, namely, cryptocurrency-directed interoperability approaches, blockchain engines, and blockchain connectors. This paper presents a comprehensive and systematic study of blockchain interoperability, mainly from the perspectives of the functional components. According to current works of literature, we classify them 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 that category and provide some discussion. As a systematization of knowledge on blockchain interoperability, we also provide some research challenges and research directions, which may help interested readers to explore more in this area.

The rest of the paper is organized as follows. Section II introduces some preliminary information on blockchains, atomic swaps, and ACID properties. Section III discusses cross-blockchain bridges. Section IV details existing solutions on blockchain interoperability for each category. Sections V presents some opportunities provided by blockchain interoperability. Section VI discusses some critical challenges in achieving blockchain interoperability. Section VII shows some potential research directions, and section VIII concludes this paper.

II. Preliminaries

This section provides the necessary background on blockchain interoperability.

A. Blockchain

1) Blockchain Basics: Blockchain is a publicly known technology underlying digital cryptocurrencies, such as Bitcoin [1]. In a broad sense, blockchain can be roughly explained as an immutable, decentralized, trusted, and distributed *ledger* based on decentralized (e.g., peer-to-peer (P2P)) networks [15]. Essentially, blockchain is a distributed data structure, and is labeled as a "*distributed ledger*" in its applications, functioning to record transactions generated within a network [16]. Typically, cryptocurrency is only one application of the functions of record-keeping, and distributed ledger technology has great potential to be adapted to other scenarios where data exchanges happen. 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 some level of security and robustness to guarantee operations on blockchain being 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 [17].

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.

2) Types of Blockchains: Depending on how blockchain organizes its participants in different application scenarios, blockchain can be roughly categorized into distinct categories, namely public (or permissionless), private (or permissioned), and consortium (or federated) blockchain [18] [19]. Each category is with distinct attributes, which will further affect the level of interoperability.

a) Public Blockchain: A public blockchain is an open and transparent network, which implies that anyone can join and participate in the consensus process, e.g., constructing and verifying blocks. Also referred to as *permissionless* blockchain, it functions in a completely distributed and decentralized way. The permissionless blockchain makes it possible for anyone to maintain an exact copy of the block data and perform the validation process on generated blocks. Typically, this type of blockchain is adopted by cryptocurrency cases, such as Bitcoin and Ethereum. A permissionless blockchain is typically designed to support a huge number of anonymous participants, so minimizing potential malicious activities is essential. Due to the anonymous participating process, some kind of "proofs" are needed to show the validity of new blocks before publishing them in a public blockchain. For example, proof could be solving a computationally intensive puzzle or staking one's cryptocurrency. Public blockchain normally requires some kind of incentive to reward the peer nodes which attempt to publish new blocks onto the blockchain (e.g., attaching a processing fee on each submitted transaction). 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 other peers are engaged in the same decentralization consensus to validate the transactions.

b) Private Blockchain: A private blockchain, on the other hand, is an invitation-only network managed by a central authority[1].

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

TABLE I. HIGH-LEVEL COMPARISON OF PUBLIC, PRIVATE AND CONSORTIUM BLOCKCHAINS

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

All participants in this blockchain must be permissioned by a validation mechanism to publish or issue transactions. This implies that any node joining a private blockchain is a known and authorized member of a single organization. Typically, a private blockchain is suitable for a single enterprise solution and is used as a distributed synchronized database designed to track information transfers between different departments or individuals. In particular, private blockchain does not need an incentive mechanism (e.g., currencies or tokens) to work, so a transaction processing fee is not needed. Note that the blocks in a private blockchain can be published and agreed on by delegated nodes within the network; hence, its tamper-resistance might not be as effective as the public blockchain.

c) Consortium Blockchain: A consortium blockchain, also known as a federated blockchain, is similar to the settings on a private blockchain, meaning the consortium blockchain requires permission to access the blockchain network. Consortium blockchains, in most cases, cover many organizations, which together maintain consistency and transparency among them. Thus, a consortium blockchain can be considered as a verifiable and reliable communication media, which is used to trace the shared and synchronized information among its participating members. The accessibility of consortium blockchain lays between the public and private blockchains, which is popular in multi-organization involved project. The consortium blockchain is very prevalent in large-scale industrial systems, in contrast to the public and private blockchains [20]. In some sense, a consortium blockchain is still one blockchain, whose collaboration within one blockchain is different from the concept of interoperability among multiple blockchains.

Based on the above discussion, Table I shows the comparison of different types of blockchains. Different application scenarios may adopt different types of blockchains. For example, a single organization may use a private blockchain, or may join a consortium blockchain as a member. And different blockchain types may affect the level and difficulty 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 (i.e., public blockchains). This will in turn affect the level of interoperability among blockchains.

B. ACID of Blockchain

Atomicity Consistency Isolation Durability (ACID) provides some general principles in database management systems (DBMS), which targets to guarantee the reliability and consistency of a given database [21]. A transaction is an instance of information exchange, which is a logical unit of work performed within a transaction processing system, e.g., blockchain.

A transaction in an ACID system should have the following features for a blockchain system [22]: (a) a 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 and valid state to another, without compromising any validation rules and data integrity constraints; (c) concurrent transactions are executed securely and independently, preventing them 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 a blockchain transaction, and also for a cross-blockchain transaction.

The work [23] proposes two distributed commit protocols, whose approaches enable non-blocking distributed commits for multi-party cross-blockchain transactions. Both protocols assume that participating blockchains either have an effective way to communicate via smart contracts or a proxy to enable communication, which focuses on a prototype design. The first one is called a synchronous cross-blockchain transactions protocol, which follows a two-phase commit protocol (2PC) and ACID properties [24], resulting in higher latency. It delays the global commit until none of the participating blockchains can unilaterally rollback the transaction. A specific blockchain, called a *coordinator*, is used to precommit messages to all blockchains and wait for replies. Each local blockchain waits for a specified amount of time before committing the message, in which the waiting time assures that the local transaction has enough confirmation time. 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 [25].

Besides ACID properties, a multi-blockchain system should follow a SALT property [26]. 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 [22]. All these features are key to successfully design interoperable blockchain systems.

C. 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) [10]. Roughly speaking, 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. Both transfers must be guaranteed to happen or neither of them happens. This property is called "atomic", as swap process is indivisable [27] [28].

The atomic swap can be adopted into multiple blockchain scenarios, which is referred to as an atomic cross-chain swap. In general, an atomic cross-chain swapping process can be considered as a distributed coordination task, which can enable the ability of multiple participants to exchange their assets across multiple blockchains atomically and collaboratively [9]. 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* [29], the whole swapping process can be executed automatically without human interventions. 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., a transaction transferring an asset from one party to another. The atomic cross-chain swap [9], as a cryptographically powered smart contract, enables peer-to-peer exchange of assets directly between two blockchains while both of them have complete control and ownership of their assets until the transaction actually happens.

An atomic cross-chain transaction is a distributed transaction that spans multiple blockchains. For example, an off-chain exchange takes place when some assets (e.g., coins or tokens) of a blockchain are exchanged for other assets hosted on another chain. Depending on where the transaction happened, atomic swaps can be classified into two major types [10]: 1) on-chain atomic swap; and 2) 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, on the other hand, takes place on a separate layer away from the chains, which can support the swapping process even among heterogeneous blockchain systems. In this case, it requires a "middleware" to facilitate the swapping process. Atomic swaps, both on-chain and off-chain solutions, bring many advantages to multiple blockchain systems, e.g., increasing interoperability and eliminating the need for intermediation.

Different schemes exist to implement atomic cross-chain transactions. One way is to make use of Hash Time-Locked Contracts (HTLC) [30]. HTLC contracts utilize time locking and pre-image revelation. They allow a party A (i.e., sending party) to first lock some assets on a blockchain such that 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 but only if 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 having to trust each other or any other third party [31] [32]. 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. The flaws of the centralized schemes are very obvious (e.g., single point of failure), and thus, achieving a peer-to-peer and trust-less atomic swap across chains is crucial to enable interoperable blockchain systems.

In practice, a different level of swapping or exchanging assets exists among blockchain users. We typically focus on transferring the ownership of assets. This means the assets are not physically transferred between different blockchains, only their ownership is transferred among the blockchains. According to different applications, this transferring process may be different in multiple blockchain systems, e.g., if the physical transfer of assets from one chain to another is required. This kind of transfer not only changes the ownership of assets across blockchains, but also changes the actual assets from one blockchain to another. In this case, the transferring process must follow the *all-or-nothing* atomic cross-chain communication protocol [33].

The technological advances of the atomic cross-chain swap are in its infant stage and still need to overcome many obstacles before being effectively implemented to multi-blockchain systems. In general, the atomic swapping process, especially in on-chain scenarios, are very slow in speed, and thus affecting the throughput of the overall system. Meanwhile, atomic swaps typically require support from smart contracts. If a blockchain system does not have the support from smart contracts, it typically very difficult to facilitate the atomic swapping process. Technically, when applying atomic swaps to the blockchain domain, it only solves the part of assets exchange problem between two entities, the need for a fully decentralized exchange is still not met, and the swap is still subject to a single point of failure. The technology on atomic cross-chain swapping is still in its infancy at this moment and the scale of the current atomic swapping scheme is pretty small. We expect the atomic cross-chain swap will likely become a fluid "background processing process" without compromising the features of blockchains [10].

D. 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 it is difficult to obtain external data, and each blockchain executes transactions on its own [34]. Cross-chain communication refers to the transferring of information between one or more blockchains. It is motivated by two basic requirements commonly found in distributed systems: accessing or exchanging data and functionality which is available in other systems [35]. 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 [14].

A typical cross-chain communication protocol refers to the procedure in which a pair of chains (including both intra- and inter-blockchain scenarios) interact in order to achieve a synchronized and consistent status among chains. An intra-chain scenario can be, for instance, a sharding blockchain, and each chain can be considered as an independent chain maintained by an independent shard. While an inter-chain scenario can consist of different blockchain systems, e.g., Bitcoin and Ethereum [36], a cross-chain communication protocol mainly targets homogeneous chains, which is a typical intra-chain scenario, e.g., Zendoo [37].

A cross-blockchain communication protocol refers to a procedure in which a pair of blockchains interact to achieve a synchronized and consistent status among blockchains [14]. It typically allows heterogeneous blockchains to communicate, which is an inter-chain

scenario, e.g., the Interledger protocol [38]. Interledger enables secure transfers on two ledgers, e.g., creating a bridge between the involved chains. In general, a cross-chain communication protocol can implement the functionalities or structures to interoperate chains within homogeneous blockchains, while a cross-blockchain communication protocol requires both source and target blockchains to follow the predefined procedure among blockchains. Both cross-chain and cross-blockchain communication protocols are important to fulfill blockchain interoperability, and they can be considered as different level protocols among multiple blockchain systems.

In general, it is more difficult to design a cross-blockchain communication protocol, since different blockchains may employ different consensus protocols, block sizes, confirmation times, hashing algorithms, and network models. In the literature, there are some theoretical claims on cross-blockchain communication. According to the well-known cross-blockchain proof problem [39], it is hard, if not impossible, 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 among them. A trusted third party, either centralized or decentralized, can help the transferring process among blockchains. This means cross-blockchain communication is not feasible in practice without the help of a trusted third party [14]. However, involving a trusted third party is against one of the blockchain features, decentralization. The cross-blockchain communication protocol requires that both source and target chains can freely exchange and verify arbitrary data information in a decentralized and trustworthy manner. And the standardization process for blockchain interoperability has a long way to go.

E. Blockchain Interoperability Definitions

The technologies to advance blockchain interoperability are still in their infancy, and no standardization efforts have gotten agreement, nor has the definition of blockchain interoperability. In this section, we provide some state-of-the-art and representative descriptions on the definition of blockchain interoperability.

Many literature papers [40] [14] mention a definition from the National Institute of Standards and Technology (NIST) (NIST Draft NISTIR 8202, Jan. 2018) 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." However, we did not find the original source of this definition from NIST. To map the blockchain interoperability into Internet infrastructure, Hardjono et al. [40] provide the definition of "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. [14] follows the source-target model and provides the definition of blockchain interoperability as "The ability of a source blockchain to change the state of a target blockchain, enabled by cross-chain or cross-blockchain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems".

While the above definitions on blockchain interoperability emphasize different domains of blockchain interoperability, the standardization of blockchain interoperability still has a long way to go. We also provide our understanding of the definition of blockchain interoperability: the ability to correctly conduct assets transferring and recording among a composition of homogeneous and heterogeneous blockchain systems, without compromising the legacy design philosophy of each blockchain system. Each blockchain system is an independent island, and interoperability is an add-on feature to interoperate these independent islands. Thus, when we add new features to blockchain systems, either homogeneous or heterogeneous, we must not compromise the blockchain's original features as a distributed and decentralized ledger system.

III. Cross-blockchain Bridges

To facilitate blockchain interoperability, a cross-blockchain "bridge" is required to enable communication from one chain to another and back, in which the bridge will function as a connector to smooth the communication among distinct blockchains. The mechanisms of atomic swaps can be implemented in the bridges. According to the properties of assets, the cross-blockchain bridges can be roughly classified into two categories: cross-blockchain token transfers and cross-blockchain smart contracts [8] [25].

A. Cross-blockchain Token Transfers

Tokens, e.g., cryptocurrencies or crypto-coins, traditionally are bounded only with one type of blockchain. Transferring tokens between distinct blockchains can be a promising research trend, e.g., from one source blockchain to another target blockchain. Token transfers between the involved blockchains should be processed in a decentralized and autonomously synchronized manner. And the token transferring process should prevent potential attacks, e.g., double-spending and the faking of transactions [41], e.g., 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 that they can get a consistent state. However, considering security, privacy, and efficiency, it is difficult and impractical to fully replicate the state of one blockchain within another blockchain to finish the verification process, without relying on a third party [7].

In general, atomic swaps can help to process the cross-blockchain token transferring process, which allows clients from different blockchain to swap their tokens atomically. And the schemes of atomic swaps have received great attention from both industry and academia to perform a cross-chain token transfer, e.g., the works [10] and [9]. Atomic swaps, on the one hand, 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. On the other hand, the atomic swapping process always requires a counter-party 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 of a blockchain. The research on cross-blockchain token transfer is still limited and in its infancy stage. Until

now, there is no practical solution to enable the token transfer among different blockchains, neither resorting to atomic swap nor relying on a trusted third party.

Atomic swaps can provide some degree of interoperability at the level of token exchange without the need for a trusted third party, e.g., a cryptocurrency exchange market. Even with several well-known proposals for atomic swaps in place, the fundamental problem of blockchain interoperability, namely, that transactions processed in one blockchain never leave that particular blockchain, is still not resolved [42]. For example, the Deterministic Cross-Blockchain Token Transfers (DeXTT) protocol [43] 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 one of the involved chains is disabled or out of service. To achieve this, the DeXTT protocol utilizes the concept of intermediaries, called witnesses, to verify and broadcast transactions to all participating blockchains. This indefinitely exacerbates the communication complexity of the whole system.

So, what does an ideal cross-blockchain token transfer look like? An ideal cross-blockchain token transfer should enable the participants to freely choose the blockchains to hold their assets, without needing some particular blockchain. This would allow the participants to keep different portions of a specific token on distinct blockchains simultanously [8], and, an off-chain participant to participate at any time without having to request permission from a centralized authority [13]. This is not an easy task to achieve for general use cases if only using atomic swaps, but smart contracts can help to leverage this situation.

B. Cross-blockchain Smart Contract

With the prevalence of applying smart contracts to blockchain applications, smart contract based cross-blockchain mechanisms have become popular. Different from cross-blockchain token transferring mechanisms, cross-blockchain smart contracts target *general* blockchain interoperability, instead of specific cases of multiple blockchain systems [8]. General blockchain interoperability aims to develop a generic communication scheme between blockchains, e.g., passing arbitrary information between blockchains in a decentralized and trustworthy manner. Thus, a generic framework that enables smart contracts on one blockchain A to communicate with smart contracts on another blockchain B and vice versa is desirable [44].

There exist several literature working on the design of cross-blockchain smart contracts. Jin et al. [45] provides an architecture for enabling interoperability amongst 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 firstly 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. Also, their work discusses the challenges of realizing interoperability in terms of atomicity, efficiency, and security. In addition, the authors state that the biggest challenge for cross-blockchain smart contracts is to connect the runtime environments of the two blockchains.

PolkaDot [46] 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 which 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. The PolkaDot whitepaper includes basic ideas on how to interact between parachains and the relay blockchain. Since only prototypes are provided and the Polkadot network has not yet to be launched, the project appears to be in its early stages of progress. Also, the Polkadot protocol supports not only token transfers, but also supports other types of blockchain interoperability [42].

Cosmos [47] 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 own blockchains such that the receiving chain may prove the packet received. Also, Cosmos requires that all zones implement the same consensus protocol to guarantee the consistency of blockchains.

Until now, the number of solutions on cross-blockchain smart contracts for generic blockchain interoperability is quite small, and no feasible solution exists in an efficient, decentralized, and trustworthy manner. The basic requirement for creating a cross-blockchain smart contract is to provide an inter-blockchain communication protocol that can be used to facilitate a decentralized and trustworthy arbitrary data exchange among blockchains. Feasible cross-blockchain smart contracts have a long way to go.

C. A Generic Cross-blockchain Protocol

This section provides a generic cross-blockchain protocol. For simplicity, we consider two independent blockchain systems X and Y, in which each works as a closed system, and we assume that a process (aka. operation) P runs on X and a process Q runs on Y. A process has the ability to affect the state of a blockchain system in two exclusive manners: (i) writing a transaction (e.g., TX) to the blockchain (commit), or (ii) stopping to interact with the blockchain system (abort). These assumptions follow the cross-chain communication system model in [48]. A generic cross-blockchain protocol consists of the following four main phases.

1) Setup: The main task of the setup phase is to exchange and parameterize the information of the involved blockchains, which is used to initialize the cross-blockchain communication so that the source and target blockchains know each other. Also, it exchanges the corresponding verification and agreement schemes, including the description summary of the transaction (e.g., designating the value of transaction and recipient). 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 two parties. In general, the setup phase happens out-of-band between the two involved parties.

2) (Pre)-Commit on X: Once the setup phase is successfully finished, a publicly verifiable commitment to execute the cross-blockchain transaction is submitted on the blockchain system X, e.g., P writes 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 X.

3) Verify: The validity of the commitment value on blockchain X by P should be verified by Q following the agreed verification scheme. And there will be two possible results: Commit on Y or Abort.

4a) Commit on Y: After successfully performed the verification on Y, a publicly verifiable commitment will be published on blockchain Y, and finally, this information will be appended in a stable block.

4b) Abort: However, if the verification process fails, or Q fails to complete the execution of the commitment on Y, the cross-blockchain protocol can then perform an abort operation on blockchain X, e.g., by "reverting" the modification to its original state. This reverting operation can be done by another transaction, e.g., blockchain X resets to the state before the pre-commit occurs.

The above generic cross-blockchain protocol can work with a two-phase commit protocol to facilitate the exchange of assets. The step 2 (pre-commit on X) is a conditional state transition, which can be reverted based on the execution on blockchain Y. Also, Zamyatin et al. [48] shows the impossibility of a cross-blockchain communication protocol without a trusted third party in fair exchange problems [49].

For a possible implementation and practical consideration, different phases may engage in different operations. For example, the commit (on phase 2 and phase 4a) typically involves a locking operation and an unlocking operation on exchanged assets of chains X and Y, respectively, according to the outcome of the actual protocol execution. 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). And the abort phase is typically an optional phase, which means once a commit is executed, no abort will be necessary.

D. Classification of Cross-blockchain Protocols

According to different rules, there exist different kinds of classification on cross-blockchain protocols, e.g., the classifications in works [13] [14]. While based on the design rationale and use cases, the cross-blockchain protocols can roughly be classified into two categories: exchange protocols and asset migration protocols [48]. The exchange protocols synchronize the exchange of assets on two blockchains, while asset migration protocols allow moving an asset or object to a different blockchain.

Typically, an exchange protocol requires an atomic swap of two (or more) digital assets, e.g., x on chain X and y on chain Y. This kind 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. For example, hashed time-locked contracts (HTLCs) (refer to Section IV) belong to this category. If we consider a two blockchain scenario, moving an asset from a source blockchain to a target blockchain, then the asset migration protocol typically is achieved by a "write block" to prevent any further updates of moved assets on the source blockchain, and to create a representation on the target blockchain. And once the assets migration is successful, the moved assets can only be operated on the target blockchain, and the source blockchain loses ownership of those assets. Typically, the crypto-currency-backed assets adopt asset migration protocols.

E. Forms of "Trust Model"

The main challenge to achieving blockchain interoperability is the trust model [50]; even in a decentralized network, trust is one of the prerequisites to conduct the interaction between blockchains. Based on the categories of cross-blockchain communication protocols, the trust model can be roughly classified into two categories: trusted third party (TTP) and synchrony [48].

1) Trusted Third Party: A TTP can be in the form of a 'coordinator' to ensure the correct execution of a cross-blockchain communication. Also, there are different criteria to classify a coordinator, e.g., custody of assets vs. involvement in blockchain consensus, and static vs. dynamic. For custody of assets, there are two forms of custody which determine control over the assets: custodians and escrows. In general, custodians have *unconditional* control over the assets and thus can be trusted to release them, while escrows have a *conditional* control over the assets according to the predefined constraints. Both types of custody are imperfect. For example, custodians may commit theft and escrows may fail to take action (i.e., freeze assets). For the involvement in blockchain consensus, there also exist two forms of coordinators: *consensus-level* coordinators and *external* coordinators, which are based on the coordinators' participationship in consensus. For the criteria of the static and dynamic, it is typically based on how the election scheme selects the coordinator. For example, the static coordinator would not be changed over time (usually in permissioned blockchain), and a dynamic coordinator can be chosen by participants of the cross-blockchain communication protocol for each individual execution.

Based on the above classifications in practice, the coordinators can be implemented in various forms. For example, external custodians can be considered in the form of committees, where the trust assumptions are literally spread among the committee members, instead of a single external coordinator. Consensus-level custodians can be considered in the form of a consensus committee (besides the roles of external custodians), which is also responsible for agreeing on the state update of the involved ledger. The external escrows can be implemented in the form of multi-signature contracts, requiring a group (e.g., a majority) of individual signatures from its committee members. And the consensus-level escrow can be implemented in the form of a smart contract, which can automatically execute and guarantee that executed results are agreed upon by its consensus participants.

2) Synchrony: Another type of trust model relies on the assumption of synchronous communication between participants and leveraging the locking mechanisms (e.g., cryptographic primitives). We can alternatively call it *lock contracts*, which facilitate asset exchange and implement two-phase commit [51]. 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 (e.g., with a strong guarantee on the message traversals) [52].

Based on different scenarios, there are also different implementations in practice, e.g., hash locks [53], signature-based locks [54], timelock puzzles [55] and verifiable delay functions (VDFs) [56]. For example, hash locks typically rely on the property of *preimage resistance* of hash functions. Signature-based locks remove the requirement that both parties' support the same hash function, which is difficult to hold in practice. Both timelock puzzles and VDFs are related to "future" activity, in which the solution to the challenges

TABLE II. 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

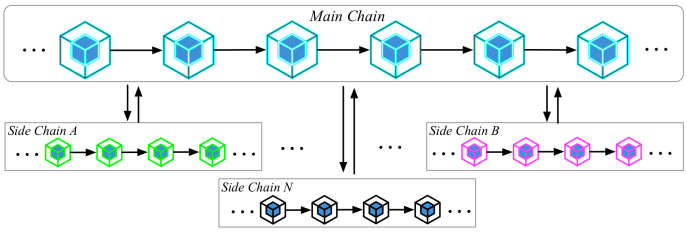

Fig. 1. Abstract of Sidechain Communication Scheme between Side chains and Main Chain

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 VDFs have more features, which are guaranteeing the validity of the released result being publicly verifiable.

Besides the above-mentioned two trusted models, there are other hybrid schemes, as in the form of *watchtowers* [57], to act as an interoperable service provider. For detailed classifications on the trust model, interested readers can refer to the Zamyatin [48].

IV. EXISTING SOLUTIONS ON BLOCKCHAIN INTEROPERABILITY

Blockchain interoperability has been a hot research topic for both academia and industry. To advance the usability of blockchain into practical applications, many different techniques and solutions have been proposed to address blockchain interoperability. According to the state-of-the-art literature and the functionalities of these solutions, we can roughly classify the researches on blockchain interoperability into three major categories: chain-based interoperability, bridge-based interoperability, and dApp-based interoperability. Each category has one or more sub-categories. We note that each category or sub-category is not disjointed, and they may overlap with each other. Table II shows a classification of blockchain inseparability solutions, and we will discuss each solution in the following section in detail.

A. Chain-based Interoperability

The chain-based blockchain interoperability mainly targets public blockchains, especially for the applications of cryptocurrencies. This category uses token swaps, such as crypto-coin swapping, as a medium to exchange information among different blockchains. Following Buterin's and Belechior's classifications on chain interoperability [13] [14], we classify three sub-categories on chain-based interoperability: sidechain, notary scheme, and hash-locks.

1) Sidechain: Sidechain is an essential innovation in blockchain, which affects the broader interoperability and scalability of blockchain networks. A sidechain can add new functionalities, namely, security and privacy, to the existing blockchains to improve their functionalities of vanilla blockchains. The initial goal of the sidechain is to extend the functionalities of interoperable blockchain networks, where data can be sent and received between the interconnected blockchain networks. This kind of design philosophy helps the security of the whole system. For instance, by isolating from the mainchain, in case of the 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 and synchronize tokens between two chains [58]. Fig. 1 shows abstract modeling of the sidechain communication scheme, with all information going through the main chain for interoperability.

The essential feature of the sidechain is to pay attention to the structure and the consensus of the chain. The mainchain generally does not in itself know the presence of the sidechain, however, the sidechain must have the ability to locate and know the presence of the mainchain [59]. Sidechains may have their own consensus protocols, which could be completely different from the mainchains' protocol. And a sidechain as a secondary blockchain connects to the main blockchain with a two-way peg [60]. A two-way peg can be considered as a scheme which enables the bi-directional assets transferring process between the mainchain and the sidechain. At the heart of any two-way peg lies a relay routine that transfers data and consensus across blockchains [61].

Schemes on Two-way Pegs

An initial design of a two-way peg is to design a systematic transfer of assets back-and-forth between consensus-disjointed blockchains. A two-way peg typically operates in some patterns: A user of the mainchain sends its tokens to a dedicated address (also known as a lock-box) where the tokens are locked, and those tokens are locked on the mainchain. The tokens can only be unlocked once tokens on the sidechain are locked and transferred back to the mainchain. After the sidechain receives the locking information on the mainchain, it creates a corresponding number of tokens. Then those tokens can be used on a sidechain by a user. Finally, the user has the ability to transfer those tokens back to the mainchain, and the corresponding assets on the sidechain are either locked or destroyed, and then an equivalent number of tokens will be unlocked on the mainchain from the lock-box [14] [60].

Currently, there exist three key options to realize a two-way peg scheme which can bi-directionally transfer assets between the mainchain to the sidechain [60], namely centralized two-way pegs, federated two-way pegs, and simplified payment verification.

a) Centralized two-way pegs: This is the simplest way to implement a two-way peg, which requires a trusted third party to hold the locked tokens. The centralized third party is responsible for the operations of locking and unlocking tokens on both the mainchain and its sidechains. While a centralized scheme provides some kind of efficiency, it is subject to a single point of failure and the centralization issue.

Advantages vs. Disadvantages of Centralized two-way pegs: Centralized two-way pegs provide two main advantages: 1) they are easy to implement and manage due to a simple design, which only involves one centralized entity to control the token transferring

process; 2) the processing speed on the token transferring process can be extremely fast as the centralized entity does not need to include a complex locking scheme.
Centralized two-way pegs have some drawbacks in the following three aspects. 1) Centralized schemes go against the decentralized design principle of blockchains, in which a two-way peg scheme introduces a certain degree of centralization. 2) The centralized two-way peg scheme will definitely introduce the chance of a single point of failure in multi-blockchain systems. 3) If the centralized entity is compromised, or behaves maliciously, it can steal all the tokens stored in the lock-box or perform any malicious operations on locked tokens.

b) 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 participants or notaries control the lock-box, instead of only one central entity. In this scheme, a set of participants collectively control the locking and unlocking operations on tokens between the mainchain and sidechain. The token transferring process occurs only if when the majority of the participants within the group sign the transferring transaction. Federated two-way pegs try to decentralize the centralized two-way pegs. A common implementation is to use multi-signature schemes [62], in which a quorum of participants signs a transaction. For example, a 'n' out of 'm' ($m >= n$) solution requires at least 'n' participants to sign the transaction to get the transaction approved. Compared with centralized two-way pegs, this achieves some degree of decentralization, but it does not completely eliminate centralization.

Advantages vs. Disadvantages of federated two-way pegs:
Federated two-way pegs provide two main advantages: 1) Compared to centralized two-way pegs, they improve the decentralization of multi-blockchain systems, and 2) they can work with some specialized federation protocols (e.g., Strong Federations [63]) for fast transfer of tokens between blockchains.
Federated two-way pegs have some drawbacks in the following two aspects. 1) Its design still resorts to a small group of participants to manipulate and monitor token transfer between blockchains, which does not completely eliminate the centralization problem. 2) Tokens in the lock-box could still have a high chance to be stolen if the majority of the participants of the federation are compromised.

c) Simplified Payment Verification(SPV): Simplified Payment Verification (SPV) [64] allows lightweight clients to verify transactions on the blockchain without having to download the full state of the blockchain (e.g., from the genesis block). The lightweight clients only need to obtain the header information of blocks, and this significantly reduces the amount of information to be downloaded. Also, the lightweight clients are required to request some proof information, e.g., in the form of a Merkle tree proof [65], to validate the target transaction is really in a valid block. An SPV two-way peg scheme works as follows: to transfer a token, such as from mainchain to sidechain, the mainchain tokens must be sent to a special address of the mainchain where only the corresponding sidechain has the ability to unlock that token by showing an SPV proof. The above process requires two waiting periods to synchronize both chains: one is the confirmation period, and the other is the contest period [66].

The confirmation period refers to a period over which a token must be kept being locked on the mainchain prior to transfer to sidechain. This confirmation period allows for sufficient work to be created. Practically, the length of the confirmation period depends on some pre-defined security parameters of sidechain, which typically requires a trade-off between cross-chain transfer 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 of this contest period is used to prevent some attacks, such as double-spending attacks. In this reorganization period, if a user finds some contradictory results to its original request, this user 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, and that proof gets approved, then this round of conversion will be retroactively invalidated. This process is typically referred to as a reorganization period. By utilizing the Merkle tree proofs, this can effectively remove the use of third parties.

Advantages vs. Disadvantages of SPV two-way pegs:
In general, one of the key advantages of this scheme is to avoid the use of the trusted third party for token transfer between the involved blockchains. The disadvantage is mainly related to a long time it takes to finish a transferring process, as a user needs to wait for the confirmation and reorganization periods before having access to the transferred tokens on either mainchain or sidechain.

Fig. 2 shows the abstract operations of the mentioned three two-way peg schemes. Fig. 2 (a) is a centralized scheme which only one central exchange entity to manage all transferring process; Fig. 2 (b) is a federated scheme, which requires multiple entities to collaborate to finish a transferring process; and Fig. 2 (c) is an SPV-based scheme, which requires a longer time to complete a transferring process.

Platforms of Sidechain

This section presents and reviews four major state-of-the-art sidechain platforms, specifically Loom [67], RootStock (RSK) [68], Liquid [63] [69], and Poof-of-Authority (PoA) networks [70].

Loom Network Loom is a decentralization Applications (dApps) platform, which runs on sidechains to connect the Ethereum, Binance Chain (living on mainnet) [71], and Tron. It is based on a federated two-way peg scheme to swap the assets among multiple chains. In its nut, Loom utilizes a Delegated Proof-of-Stake (DPoS) protocol [72] to get agreement, and each dApp can independently run atop its own sidechain (called a DAppChain) which then is being pegged to the underlying Ethereum mainchain. Along with the DPoS consensus, Loom also runs on a Byzantine Fault Tolerant (BFT) consensus [73] as a backend P2P layer (called Tendermint [74]). A transaction on the Loom network is not immediately settled on the Ethereum mainchain; instead, it is settled in bulk. According to the Loom whitepaper, Loom allows for any consensus mechanism to be implemented on a personalized sidechain (DAppChain).

RootStock (RSK) Network RSK is a general-purpose smart contract platform where sidechains are pegged to the Bitcoin mainchain. RSK utilizes the scheme of merged mining [75] to provide incentives to the miners who are actively involved in the mining process on the RSK platform. To improve mining efficiency, RSK utilizes DECOR+ [68] protocol, a reward-sharing scheme to reduce competition while mining, and DECOR+ can deterministically resolve the conflicts since all nodes finally will get the same information on-chain state. RSK allows users to mine in both RSK and Bitcoin networks without performance penalties.

RSK relies on a combination of a federated two-way peg with an SPV scheme. For the asset transfer, a token of 'SmartBitcoins(SBTC)' is used to transfer, e.g., from Bitcoin blockchain to the RSK sidechain, and the SBTC is essentially a Bitcoin natively on the RSK platform,

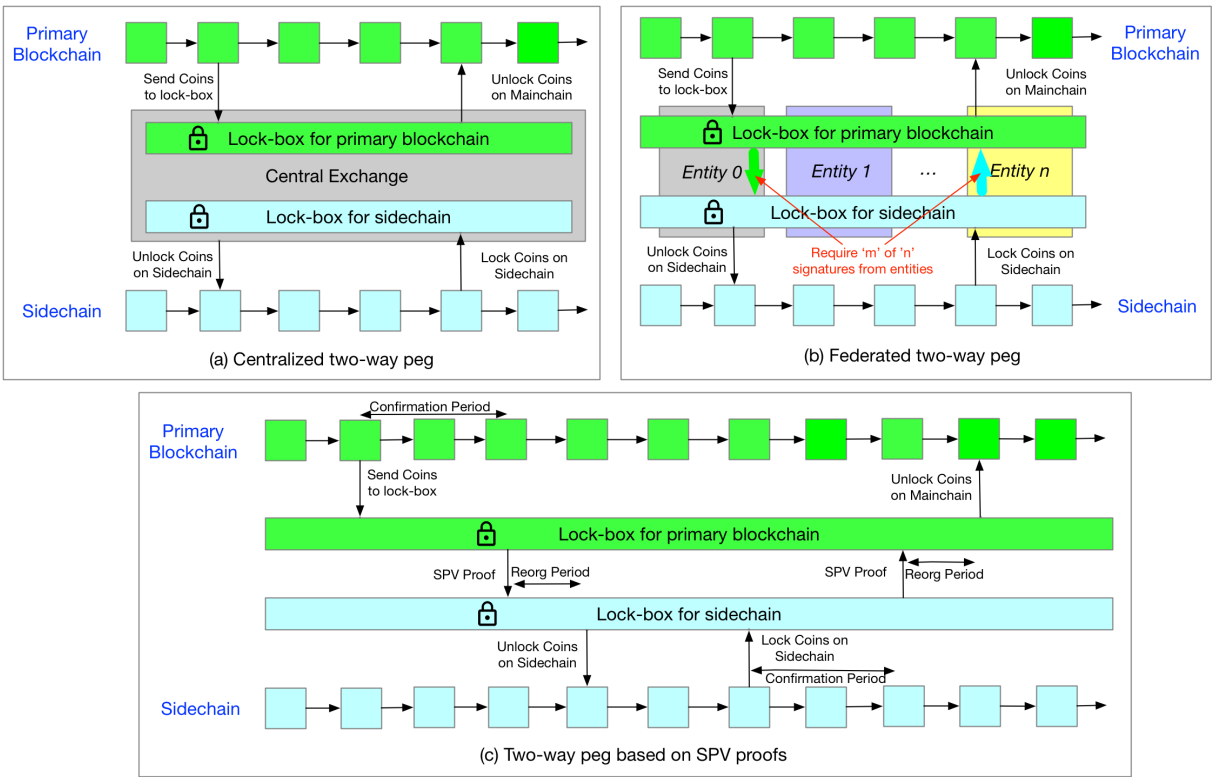

Fig. 2. Two-way Peg Schemes: (a) Centralized two-way peg, (b) Federated two-way peg, and (c) SPV-based two-way peg.

and this platform can transfer the coins back to the Bitcoin network at some specified time with a standard RSK transaction fee. Due to the federated two-way pegs, each transfer requires a multi-signature to finish the transferring process, where the multi-signature is controlled by the RSK Federation (e.g., several stakeholders). Federation members use hardware security modules to protect their private keys and enforce the underlying transaction validation protocol.

Liquid Network Liquid is a federated two-way pegged sidechain, relying on the concept of 'strong federation'. Originally, strong federations were designed to solve the problems related to transaction latency, commercial, privacy, reliability, and fungibility, in which several financial institutions and cryptocurrency exchange platforms to run high-performance computing hardware to secure the network. A strong Federation scheme consists of two independent types of entities, namely block-signers and watchmen. Block-signers maintain the blockchain consensus and advance the sidechain, while watchmen realize the cross-chain transactions which are responsible for transferring assets from the sidechain to the mainchain by signing transactions on the mainchain. Liquid network also utilizes a multi-signature scheme to sign each block transferred between mainchain and sidechains. Liquid network supports multiple types of assets, such as traditional currencies, real-world assets, and other cryptocurrencies in addition to Bitcoin.

Proof-of-Authority (PoA) network PoA network is an Ethereum-based sidechain platform. The PoA network is intended to allow a cross-chain transferring process between Ethereum to a side chain with more scalability and interoperability between other blockchain networks. It also provides bridging capabilities which allow users to transfer their non-fungible tokens from one blockchain to another easily, which provides a solution to communicating between two arbitrary stand-alone blockchains. 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 amount staked. Also, the PoA network provides different types of asset transfers, e..g, Native (i.e., PoA tokens) to ERC 20 [76], ERC20 to ERC20, and ERC20 to Native.

Besides the above four main sidechain projects, there exist several ongoing projects to realize blockchain interoperability by using sidechain solutions. For example, Plasma [77] aims to provide a highly scalable solution for the blockchain-based decentralized financial industry. Blocknet [78] is a PoS-based platform, which consists of XBridge, XRouter, and XCloud, and XBridges relies on SPV for a two-way pegging process.

Open Issues on Sidechain Solutions

Sidechains are still a relatively new proposal to deal with blockchain interoperability. Although these solutions are promising for the future of the blockchain industry, they also come with some open issues.

Centralization Issue A two-way peg sidechain, either centralized or federated, is subject to the centralization issue. It is clear for the centralized two-way pegs, and federated two-way pegs introduce a certain level of political centralization. For a federated two-way peg, it is critical to choose some honest and trustworthy individuals to form a federation for the purpose of security and consistency of the blockchain ecosystems. However, forming a "good" federation is not an easy task (even resorting to randomized schemes), and a "good" solution has to have a majority of federation members who are honest and trusted. A good federation has to have some properties, e.g., the

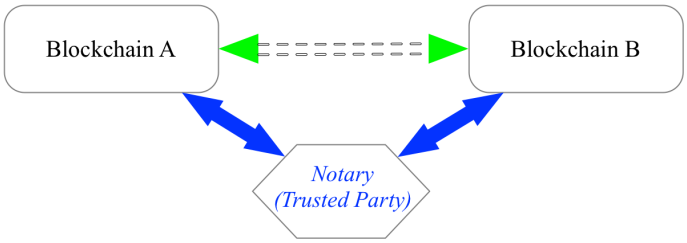

Fig. 3. Abstract of Notary Scheme via a Centralized Trusted Party as a Relay

identity and authenticity of each individual should be verifiable and individuals should be distributed geographically. Also, the size of a federation (i.e., the total number of individuals in a federation) is important. If the size of a federation is large, the verification process will be relatively long, while if the size of the federation is small, the level of security of the federation might be an issue. Meanwhile, a federation must use a consensus protocol to get agreement among all honest members.

Complexity The decentralization of an SPV based design is due to the fact that the lockbox on both the mainchain and the sidechain is controlled by the participants of the network. Though SPV-based two-way pegs do not have a centralization issue, they introduce additional complexity on different levels. For example, on the network level, the participants must maintain and cooperate with multiple independent unsynchronized blockchains supporting transfers between each other. It is required that transaction scripts can be invalidated by the participants if there exists a later reorganization proof. On the assets level, it is no longer a simple assumption "one chain, one asset", and individual chains may support arbitrarily many assets. This creates great difficulties in the verification process.

Besides the above two obvious open issues, there are other issues, such as the security of federated two-way pegs, and the soft-fork on SPV-based designs. Federated two-way pegs schemes require a majority of honest participants (e.g., 'n' out of 'm') 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 unsyncrhonization among various sidechains [79].

2) Notary Scheme: The naive idea behind a notary scheme is to have a trusted witness to the ownership or ratification of a contract among mutually untrusted parties. The servers provided by a notary are required to prove the existence and the ownership of a given asset at a given time. The immutability and timestamping properties of blockchains allow storing information at a certain time, which can not be modified in the future. It is natural to use the blockchain as a decentralized notary system. The blockchain notary schemes can provide the functionalities of timed proof of existence, whose proof can be used as further proof of ownership [80]. Each blockchain notary scheme can be considered as an independent blockchain system. Multiple application scenarios may exist in different blockchain notary systems. And communication such as cross-systems operations between these systems are often required. We can re-use the notary scheme again to monitor and facilitate activities among blockchain systems.

The technologically simplest way to achieve cross-chain operations is via the use of notary schemes. In a notary scheme, a notary is a trusted individual or a group of individual that monitors and manages multiple chains, initiating transactions in a chain upon the occurrence of some valid event or a particular request (e.g., via the deployed smart contracts) which typically happens on another chain [14]. A trusted individual or group of individuals (as the notary) is used to claim to one chain (e.g., Chain A) that certain information in another chain (e.g., Chain B) is valid. It typically requires a subset of trustable servers [81]. A notary monitors newly submitted activity, and checks the validity of the activity. In general, a notary serves as the mediator of the transaction between blockchains. Roughly speaking, a notary scheme is much like a centralized scheme, and there is not too much literature on this topic. Notary schemes can facilitate most cross-chain operations and are relatively simple. However, a set of notaries can be used to decentralize the consensus process among the notaries. Thus, the notary scheme may have different degrees of decentralization [13]. Fig. 3 shows a conceptual communication scheme between two blockchains, and all the information will be recorded in the notary or trusted party.

Notarization is a way to prevent fraud and guarantees the parties that a transaction is genuine and can be trusted. In a notary scheme, the cross-chain transactions highly depend on a third-part notary. Technically, it is easy to implement a notary scheme, whose security in the scheme highly depends on the reputation and honesty of the notaries. And most notaries are trusted anchors, e.g., centralized third parties. These schemes are much like centralized exchanges and banks [82]. Different from sidechains, notary schemes are like a third-party like 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 cryptocurrency exchange platform to allow the assets (i.e., various crypto-currencies) to exchange with the guarantee from the platform provider.

Notary schemes utilize the third trusted entity as the intermediary between blockchains. The role of the notary is thus to verify the correctness and integrity of information transferred to guarantee consistency among blockchains. One major advantage of the notary scheme is that it is simple, as no additional changes are required in the underlying blockchains. The trustworthiness of blockchain transactions is assigned to the notaries. For example, one potential scheme is that both involved individuals select a group of notaries that they trusted [28]. Also, the output of these notaries can get an agreement with the help of the consensus protocols, e.g., Byzantine Fault Tolerant (BFT) protocols [83]. There is no need to trust every individual notary, but only two-thirds of the set of notaries [81].

Herdius [84] is a decentralized exchange platform using a notary scheme, with its focus on some common connection points between blockchains. The notaries in Herdius are called "assembler nodes", and each one holds sliced and distributed keys for the involved blockchains. It features the solution of using threshold multi-signature schemes. Multiple assemblers have the ability to sign a transaction by using some known threshold signature scheme, and no assembler has the ability to individually decode the native private key without the help from other assemblers (e.g., a majority of assemblers). By integrating the threshold scheme, Herdius aims to partially decentralize the notary scheme. Bifrost [5] is another project that employs a notary scheme, 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

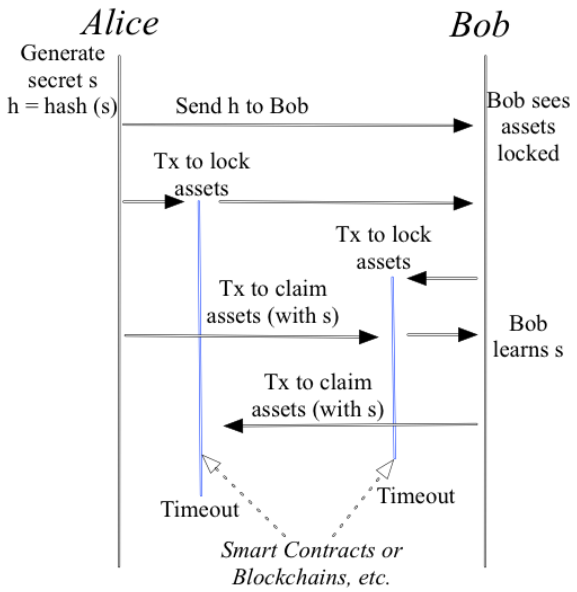

Fig. 4. Abstract of Hash-locking Communication Scheme between Two Clients using Different Blockchains [13]

honesty of the notary nodes. Besides these schemes, there exist other schemes which do not explicitly use the notary scheme to achieve interoperability. For example, Interledger [85] combines sidechains with notary schemes, and AION [86] provides a prototype that relies on a notary scheme to create an interoperable network.

One of the key benefits of blockchain is the removal of trust between participants to achieve decentralization. If this property could hold for notary schemes, it would be beneficial and be easy to achieve. But having a notary scheme means that the notary is trusted by a user. The notary has access to the private keys which makes him vulnerable to attackers. Besides, the notary controls the applications and the node, and has the ability to arbitrarily alter the original transactions, e.g., altering the data of transactions. This may lead to the blockchain applications being never trust-free, and requires other layers to guarantee the trust among blockchains [28]. In summary, while notary schemes take advantage of their atomic process, ease of implementation, supporting capabilities for different blockchain networks, nevertheless, there also exist several main drawbacks, e.g., inefficiency, lack of flexibility, and the risk of centralization. In the literature, there is no notary scheme available to handle these drawbacks.

Practically, a way to enable interoperability using a notary scheme is to combine other techniques, e.g., the sidechain technique, with decentralized forms of a notary scheme, in which a group of notaries agree on the transactions. For instance, we can integrate a federated two-way peg scheme to a notary scheme, to lease the issue of centralization. However, as blockchain interoperability is still in its infancy, it is likely that upcoming new technologies will achieve full decentralization and make blockchain interoperability more accessible for the masses.

3) Hash-locking: Hash-locking is another technique for the exchange of assets without a trusted third party [87]. Roughly speaking, the hash-locking technique uses a hash time-locked system, applying a *time lock* to lock the transaction. Only when both involved parties are agreed upon the obligations, the transaction would keep in a locked state, which is similar to the concept of an atomic transaction [9] [13]. In general, the atomic assets exchange in the hash locking scheme is achieved through time difference and hash operation. For example, AltChain provides a technical prototype of a hash lock for an atomic transferring process [88]. 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, unlocked execution, and ensures the atomicity of cross-chain transactions [89]. Some works on off-chain payment networks and state channels use a hashed time-lock contract, a kind of smart contract, to circumvent the scalability limits of existing blockchains. For example, Herlihy et al. [90] constructs a complex distributed computing platform to manage cross-chain asset transactions with a time-lock commit protocol.

It is necessary to know how the hash-locking scheme works. We assume the sender Alice wants to pay Bob, with a lock time t, and hash function $hash$. The basic principle of hash-locking works as follows [13]: 1). Alice generates a random secret, e.g., s, and computes a hash value, e.g., $h = hash(s)$, then, sends the hash value h to Bob; 2). Alice and Bob both are required to lock their assets into a smart contract with some 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 in order to claim the asset from Bob's contract. The above steps are provably atomic. For details, the interested reader can refer to the R3 work [13]. Fig. 4 shows an abstract of a hash-locking scheme to perform an atomic swap between Alice and Bob, which connects different blockchains. Locking the assets can happen on smart contracts.

Another concept in hash-locking is hashed time-locks contracts (HTLCs), which was proposed to enable cross-chain atomic operations [91]. In an HTLC, a client commits to making the transaction by providing cryptographic proof before a timeout to the other. It is typically used in Payment Channel Networks (PCNs) [92]. A payment channel establishes a private peer-to-peer medium, ruled by a set of pre-set instructions, e.g., smart contracts, which allow the involved participants to consent to the state updates by exchanging authenticated state transitions off-chain [93] [94]. HTLC is proposed to be used in PNCs to avoiding setting up payment channels, while still preserving high transaction throughput. Typically, every participant in HTLC will associate with a maximum time frame that they can pull the payment from the sender to avoid any part suspending the channel by refusing to forward the payment.

However, the hash-locking solution is not a one-shot solution, and it has some drawbacks. For instance, a hash-locking solution must lock some assets during its opening phase for an established transaction channel, however, there is a chance of asset loss if a timeout occurs. This also will create a race condition. For example, malicious participants can issue many fake transactions to block the normal communication channel, and this will significantly affect other legitimate transactions by honest participants [94].

Besides the use cases in the lightning networks, several literatures explore the usability of hash-locking for blockchain interoperability. *CheaPay* [95] targets a payment channel network, which offers the off-chain settlement of transactions between blockchains. Within a PCN network, it utilizes an HTLC scheme to guarantee the atomic swap of assets. Chameleon Hash Time-Lock Contract (CHTLC) [91]

also targets on PCN network to provide the privacy-persevering service among assets 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 [96] is a protocol stack that facilitates atomic swaps based on HTLCs. It provides two kinds of protocols, one is the cryptographic protocol, and the other is the communication protocol. The cryptographic protocol defines the order and semantics of interactions with ledgers, and the communication protocol defines the way that two COMIT participants interact to perform an atomic swap. It also provides several specific tokens, e.g., the HAN (HTLCs for Assets that are Native to the ledger), the HErc20 (HTLCs for the Erc20 asset), and the HALight (HTLCs for Assets on the Lightning ledger), to support direct assets exchange. Anonymous multi-hop locks (AMHLs) [97] define a cryptographic primitive which functions as a cornerstone for the secure and privacy-preserving PCNs for both scalability and interoperability. It utilizes several provably secure cryptographic instantiations which in turn make AMHLs scheme compatible with the current major cryptocurrencies. In the part of the token transfer, it utilizes a hash-locking scheme to guarantee atomicity. Sprites [98] also 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 technique, in which a global contract called PreimageManager(PM) is created to manage the payment transactions. Atomic Loans [99] enables the transfer of value 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.

We need to notice that all presented schemes are not orthogonal, and different schemes can work together to provide chain-based interoperability.

4) Discussion of Chain-based Interoperability: Protocols on chain-based interoperability are still relatively new, and many projects on chain-based interoperability are still in development, and some of them are still in conceptual and prototype design. We provide a brief discussion on the above three major chain-based interoperability issues.

Theoretically, sidechain solutions not only can provide interoperability among multiple chains, but also increase the scalability of the mainchain network, e.g., by performing transaction processing and verification before submitting to the mainchain. In general, sidechain-based projects target permissioned blockchains which only authorized participants can join, thus improving scalability and feature development. Different organizations can develop their own sidechain (e.g., as an atomic zone), via the mainchain to communicate. From this perspective, sidechain solutions provide a kind of isolation in nature, and this can potentially protect the whole system. For example, a sidechain may be compromised by an attacker, however, by isolation, it provides a barrier to propagate the malicious behaviors to the mainchain.

However, the sidechain solution is not a perfect solution to deal with blockchain interoperability. The first issue in the sidechain is centralization. Centralized or federated two-way pegs in the sidechain solution introduce a level of centralization; a federated scheme, for example, it is critical to choose honest and trustworthy entities as its members. The integrity and security of these systems highly rely on the honesty and security of the federation. For example, if a malicious group of entities forms a majority within a federation, it may introduce a security flaw, e.g., the assets can be locked in the lock-box. In general, SPV provides a solution for the centralization issue in federated two-way pegs, in which SPV does not require a single entity or group of entities for transferring assets between the mainchain and the sidechain. However, the SPV solution requires a long verification process to finish a transferring process as the node needs to wait for confirmation and reorganization periods.

Besides, sidechains introduce additional complexity on several levels [66]. On the network level, multiple independent unsynchronized blockchains are required to support asset transfers among each other. This means each independent blockchain must support transaction scripts. On the assets level, there is no simple principle that "one chain, one asset". Individual chains typically carry multiple assets, including those not existed when the chain was first created. Also, there may exist fraudulent transfers that are hard to detect on the target blockchain.

The notary scheme, on the other hand, is an intermediary to validate or execute blockchain transactions [28]. Because of independence on the changes of blockchain's implementation, this solution works for all current blockchains and is comparably easy to implement. However, centralization is an issue in nature. To correctly process a transaction, a notary must behave honestly, e.g., no censor and no alternation on a transaction. Also, centralization issues make the notary a target to attack, which in turn is subject to a single point of failure. In general, the use of notary schemes weakens the feature of decentralization and trustworthiness. However, no blockchain applications are completely trustworthy, as there is always some layer where trust is involved.

In general, hash-locking solutions can allow asset exchange in a trustless way from the chain level, by performing atomic swaps between different blockchains. However, since the hash-locking scheme is based on the lock and unlock of assets, assets may be lost due to the timeout. Also, for each atomic swap, the multiple transactions may yield a long waiting time.

Generally, a practical way to enable chain-based interoperability with current major blockchain systems is to combine these solutions together to form a robust multi-chain assets transferring system, e.g., combining the sidechain technique with decentralized forms of a notary scheme.

B. Bridge-based Interoperability

Bridge-based interoperability targets the implementation of a "bridge" as a connection component between homogeneous and heterogeneous blockchains. Most existing chain-based blockchains are homogeneous blockchain systems, in which the assets transferred or exchanged are of the same or similar type. We still need a facility, as a bridge, to interconnect heterogeneous blockchain systems and cross-blockchain communication. We categorize them into two major types: trusted relay, and blockchain engines. The bridge sits in the middle of communicated blockchains to maintain the integrity and consistency of each involved blockchain.

1) Trusted Relay: Trusted relay is a more "naive" approach for facilitating interoperability, where trusted parties redirect transactions from one blockchain to another. Typically, a relay enables the recipient chains to verify activities that happened in other chains.

Essentially, we can consider a trusted relay as a 'bridge' which is used to provide smart contract service between blockchains. Different from notary schemes, trusted relays operate at a chain-to-chain level without the trustworthiness on distributed participating nodes. In such a way, relays enable a contract of one chain working like a "client" of another chain. Typically, relay schemes replicate block information of the source blockchain, e.g., 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 [13] [100]. 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 the cross-chain communication in a trusted manner.

There exist many relay schemes, e.g., BTC Relay [101] and PeaceRelay [102], which utilize SPV scheme to verify transactions across blockchains. These relays are essentially SPV clients for a source blockchain, which runs on a target blockchain. For example, BTC Relay is a relay running on an Ethereum blockchain (target blockchain), which includes the transactions that happened on the Bitcoin blockchain (source blockchain). The relay needs to know the block information on the source blockchain for successful verification; then the target blockchain utilizes SPV to verify particular transactions that happened on the source blockchain. Different from notary schemes, relays are typically distributed in a decentralized manner, in which no centralized entity exists. 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. Existing relays, e.g., BTC Relay, typically only perform the verification of the source blockchain's header for every submitted block. And typically, performing SPV validation for *every* block header of the source blockchain also leads to extremely high operational cost. PeaceRelay is a relay for Ethereum-based blockchains, which requires authorized clients to submit block headers for verification. Without on-chain validation, centralization issues still exist. In the following part, we briefly introduce several well-known trusted relay-based schemes, although some of them are not fully decentralized schemes.

a) Hyperledger Cactus: Cactus is a part of the Hyperledger project, which aims to provide a secure, decentralized, and reliable platform among distinct blockchains [103]. Originally, Hyperledger used a trusted escrow to validate cross-chain transactions, known as Blockchain Integration Framework (BIF), in which it provides an integrated service to manage multiple blockchains and execute some pre-defined interoperable operations across them. Essentially, the BIF is a centralized scheme. Current Cactus targets decentralized validations, moving towards a decentralized trusted relay scheme. However, the current version is still not a fully decentralized scheme. Similar to the notary scheme, the current Cactus still requires a party or a set of coordinated parties to perform the verification.

In a nutshell, Cactus uses a set of interoperable validators to verify the cross-chain transactions, and these validators are responsible for signing and delivering the cross-chain transactions. Similarly to Hyperledger Fabric [104], the trusted relays together form a membership service provider (MSP), and validators can be in one of the "member" types. Typically, the given transactions (e.g., submitted from clients) must be signed by a quorum of validators to make them valid. Current Cactus has several transferring patterns: value transfer, value-data transfer, data-value transfer, data transfer, and data merge. Besides, Cactus provides multiple use case scenarios via a trusted consortium, where trusted relays allow discovery of the target blockchain. Operations are controlled and carried out under the Hyperledger Cactus Business Logic plugin, and this module typically is offered by vendors.

b) Testimonium: Testimonium [100] is a blockchain relay scheme that relies on a scheme, call validation-on-demand pattern, and the on-chain execution of SPV scheme to allow verification of data across blockchains without sacrificing the fully decentralized feature. It targets developing an atomic-commit mechanism for distributed transactions between multiple blockchains. Testimonium requires no trust in a single entity as validations are executed on-chain with a reward structure incentivizing participation. In general, the Testimonium scheme consists of relays (running on 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 responsible for detecting and disputing submitted illegal block headers.

Besides trusted relay projects, there exist other prototype projects on this topic. For example, an escrow-based transfer protocol prototype is proposed, called interactive multiple blockchain architecture [105], 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. Smart Contract Invocation Protocol (SCIP) [106] is another protocol prototype that provides a uniform integration for both homogeneous and heterogeneous smart contracts across 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. And SCIP prototype can be implemented at the gateway to coordinate the cross-chain transactions via smart contracts.

c) Tesseract: Tesseract [107] is a real-time cryptocurrency exchange protocol using trusted hardware as a trusted relay. It supports some secure assets tokenization scheme which can peg these assets to cryptocurrencies. For instance, Tesseract-tokenized bitcoins can be used in Ethereum chain taking the advantages of smart contracts, e.g., without relying on a human element for security. To achieve that property, Tesseract supports cross-chain trading with the help of a trusted execution environment (TEE), which behaves like a trusted third party to control funds without exposing them to theft. 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 settlement protocol to achieve an *all-or-nothing* settlement. However, this cryptocurrency exchange protocol only targets the cryptocurrency exchange, and the protocol is still in prototype design.

Tesseract assumes a network adversary, potentially the exchange operator, who can gain full physical access to the host application, and provide full control over the operating system and network connection. However, it does assume that the adversary can neither observe nor can tamper with the code running inside the TEE enclave. The enclave allows remote users for remote attestation which can guarantee the trustworthiness of the communication. Besides, it also targets providing real-time service by utilizing a simplified verification scheme.

In general, trusted relay schemes are highly usable and reliable,

with the features of asset portability, atomic swaps, and applicable for complex use cases without clear restriction [13]. However, fully decentralized trusted relay networks still have a long way to achieve.

2) Blockchain Engines: A blockchain engine typically requires a shared infrastructure to support different layer services, including network, consensus, incentive, etc., and the shared infrastructure provides a kind of "relay" among blockchains. Due to the requirements of multi-layer supports, most existing blockchain engine based solutions are still in the stage of proof of concept or under development. However, there do exist several projects that are in progress, e.g., Polkadot [46] [108], Cosmos [109], WanChain [110], and ARK [111].

a) Polkadot: Polkadot aims to provide interoperable blockchain networks among heterogeneous multi-chains, which allows the interoperability among many distinct blockchain systems and even each with various consensus protocols. Also, with the help of fully decentralized "federation", Polkadot enables different types of blockchain systems, either open or closed, to access each other in a trust-free manner. Its underneath offers a "relay-chain" which can support many validatable and globally consistent dynamic data structures (called parachains or parallelized chains) in a side-by-side manner. Each parachain can be considered as an independent chain. In Polkadot, there typically are four basic participants, namely *validators*, *Nominators*, *Collators*, and fishermen, and each has its own roles and functions. Validators typically are used to help sealing new blocks on the Polkadot system, whose key role is to ensure the contingency upon enough high volume bond being deposited. A validator has to run a client implementation of the relay-chain with high availability and enough communication bandwidth, whose process involves receiving, validating, and re-publishing candidate blocks. If a validator is provably not fulfilling its role, it will be slashed, e.g., part or all of its bond will be taken. The role of validators is equivalent to the miners in mining pools on current PoW-based blockchains. The role of nominators functions as a state-holding party who contributes to a security bond for a validator. The role of collators is to assist validators in producing valid parachain blocks by holding a full status for a particular parachain. The fishermen typically are not required to directly engage in the block-authoring process, which functions as a bounty hunter to discover misbehaviors. They also can get rewards for detecting misbehavior and function to ratify invalid parachain blocks.

The selection of Polkadot validator is based on the Nominated Proof-of-Stake (NPoS), which can achieve high levels of security and scalability. From a security perspective, Polkadot uses the Byzantine Fault Tolerant protocol to get a consensus for newly generated blocks among validators. The validators are then distributed to distinct rotating subsets, and one for each parachain to attest the validity of parachain blocks. To achieve affordable scalability, due to the BFT protocol, the size of the rotating subsets must be small enough. Cross-chain Message Passing (XCMP) protocols are used to send messages to each other among parachains. The adopted XCMP protocol has several unique features: 1) messages arrive quickly, 2) messages follow in order, 3)arriving messages are indeed sent in the finalized history of a sending chain, and 4) recipients should receive messages fairly (among senders) to ensure each sender will not wait indefinitely before their messages being seen. Besides, the relay chain of Polkadot achieves consensus using Blind Assignment for Blockchain Extension (BABE) and GRANDPA protocols [112].

b) Cosmos: Cosmos is a multi-chain system similar to Polkadot, designed to solve blockchain interoperability. Each independent parallel blockchain is called a zone (sometimes referred to as a "shard"), which is essentially a Tendermint blockchain [74]. Zones in Cosmos are the blockchains that can plug into the network for data exchange between them. Zones can transfer data to each other via an entity called *Hub*, which can minimize the number of connections between zones. The Cosmos Hub is a blockchain connector that can support a multi-asset distributed ledger, where tokens are typically held by individual zones. The Hubs connect all of the zones, acting as a localized coordinator 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.

The token exchanged is via a special inter-blockchain communication packet, and the hub is in charge of maintaining the global invariant of the summary information of each cross-zone token. IBC protocol 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. In general, Cosmos consists of three layers: 1) bottom - Temdermint, 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 kinds of blockchain can also interoperate into Cosmos networks, e.g., via peg zones. The concept of peg zones is much like a pegged sidechain scheme. Also, there is a useful socket protocol, Application Blockchain Interface (ABCI), connecting the Tendermint engine to the application, and ABCI can be easily embedded into the most existing programming language. Furthermore, the Cosmos SDK offers a generalized architecture for developing stable blockchain frameworks on top of Tendermint BFT.

The cross-chain operation provided by the Cosmos zone, on the other hand, highly relies on the feature of instant finality on 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. According to the whitepaper of Cosmos, it aims to provide an 'Internet of Blockchains (IoB)', offering a decentralized communication network among blockchains.

c) WanChain: WanChain allows interoperability between various heterogeneous blockchains, currently focused on cryptocurrencies-based blockchains, aiming to offer an infrastructure for cross-chain operations between distinct blockchain networks. For example, Wanchain can support cross-chain operations among chains (e.g., either public or private, or both). In a nutshell, Wanchain adopts a PoS consensus protocol to get a 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. From a high-level perspective, the communication of cross-chain protocol includes three key modules: the registration module (e.g., registering the original chain participating in cross-chain transactions, and registering the asset to be transferred), the cross-chain transaction data transmission module (e.g., making transaction requests, and acknowledging the receipt of the transaction), and the transaction status query module (e.g., providing the querying service on the confirmation status of the involved asset). 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) [110].

The Wanchain project announced the release of the T-Bridge (Trusted-Bridge) framework to enable universal blockchain connections. T-Bridge refers to a framework with modular components and protocols which is consistent with the original Storemen cross-chain scheme (which currently is considered as public-to-public blockchain connections). To allow cross-chain transactions, the T-Bridge model connects components from the source chain, target chain, and routing chain together. Via smart contracts deployed in blockchains, T-Bridge enables both users and service providers from different blockchains to perform cross-chain operations. Besides, WanChain also requires two components: an intermediary router chain to register and synchronize information, and protocol mediators to monitor the state information of different chains [113].

d) ARK: ARK [111] project tries to provide a platform for blockchain interoperability, which mostly relies on the concept of the bridge. According to specific needs, ARK builds bridgechains, either interlink or work independently, which enable users to exchange data and build specific use cases. ARK offers interoperability via a multi-chain approach and ARK SmartBridge technology, in which complex processes are executed on bridgechain and only the execution results will be transferred back to the main chain. The ARK's public network (or ARK mainnet) provides a platform for other blockchains to exchange assets, and newly created chains can connect to the ARK miannet using ARK SmartBridge. The ARK mainnet only supports and performs a set of specialized functions, and this enables the ARK mainnet to serve as the hub of the overall ARK system.

ARK's SmartBridge defines two types of communication protocols: Protocol-Specific SmartBridge and Protocol-Agnostic SmartBridge (or Protocol-Independent SmartBridge). Protocol-Specific SmartBridge refers to a communication layer targeting ARK-based application-centric blockchains, which mainly operates within the ARK network of bridgechains. Protocol-Agnostic SmartBridge aims to connect blockchains that adopt different consensus protocols, which mainly are used for cross-chain communication. For newly created bridgechains, a bridgechain registration process is needed to connect to ARK mainnet. The ARK public network acts as a proxy and decentralized guard for interchain communication. The ARK public network 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 ARK.

ARK project also enables "ARK-ANY SmartBridge" mechanisms, called ARK Contract Execution Services (ACES). ACES demonstrates a two-way transfer between ARK and other legacy crypto-currencies, such as Bitcoin and Ethereum, regardless of the underlying protocols. However, ACES is not a fully decentralized solution, as intermediary nodes are required to achieve interoperability.

3) Discussion on Bridge-based Interoperability: Bridge-based interoperability solutions are typically used to deal with heterogeneous multi-blockchain systems, in which each blockchain has its own chain structure, verification mechanism, consensus protocol, and smart contract, etc. The use of a bridge can be as a channel or connector to get rid of the incompatibilities, and leads the cross-chain communication manageable. If we compare it with the Internet protocol, the bridges function as routers, where outputs from a source blockchain network are processed and transferred to the inputs of another blockchain network. In general, trusted relays are much simplified and easily adopted solutions to handle interoperability, having a certain degree of centralization, which utilizes a mechanism similar to the notary schemes to interact with another blockchain. Blockchain engines are very recent solutions (e.g., Polkadot, Cosmos, and ARK have been launched and released within the recent two years), which utilize a mechanism similar to the two-way pegged sidechains or hash-locking solutions to interact with other blockchains.

Bridge-based blockchain solutions may adopt different cross-chain communication protocols, e.g., Polkadot uses cross-chain message passing (XCMP), Cosmos uses inter-blockchain communication (IBC) protocols. The cross-chain communication scheme is highly related to their overall architecture design, e.g., the different roles of nodes. No existing communication protocol is perfect, and each communication protocol has its advantages and disadvantages. For example, IBC is a more generic solution than XCMP, which can allow users to customize their zones and provide more freedom on security and validation, while XCMP restricts these customizations, but offers a more secure framework for communication (via a shared security layer). Also, bridge-based solutions typically support the extension of smart contracts, in which the developers can design and deploy their own smart contracts.

Bridge-based blockchain solutions provide convenience for end-users, and the end-users do not need to know what happened in the "bridge". It is much like an Internet Protocol (IP) in an Internet protocol suite, and end-users only need to send these "packets" (e.g., cross-chain transactions) to the bridge, and the bridge will do some transformation and processing. Different from IP protocol, bridge-based blockchain solutions do not interoperate with each other, and they require specific bridges to handle the communication among heterogeneous blockchains. End-users are responsible to choose between the existing solutions. Besides, bridge-based blockchains typically require a transaction fee to keep the network operating fairly and this may further limit the use of an interoperable blockchain construction. It would be desirable to define a standardized bridge solution, e.g., via international organizations, and different blockchain networks that can work smoothly on a standardized bridge solution. However, the standardization processes of blockchain interoperability still have a long way to go.

C. dApp-based Interoperability

One of the main goals of blockchain is to apply it to various applications, benefiting from its decentralization, immutability, and trustworthiness. Typically, we call these kinds of applications (by integrating blockchain) decentralized applications (*dApp*, *Dapp*, or *DApp*), which are distributed Internet applications operating on a decentralized P2P network (blockchain). Similar to other blockchain solutions, dApp-based blockchain solutions also need to be interoperable. However, dApp alone cannot ensure semantic interoperability. Thus, it is essential to ensure that a dApp supports minimum structural interoperability and potentially achieves interoperability among dApps. This section discusses some approaches to achieve dApp-based blockchain interoperability, especially when we classify them into three main categories: blockchain of blockchains, blockchain adapters, and blockchain agnostic protocols.

1) Blockchain of Blockchains: The blockchain of blockchains (BoB) provides a platform for developers to construct cross-chain dApps, and each blockchain functions as an independent blockchain. For ease of expression, we can consider the top-level blockchain the

mainchain, and each other blockchain in the BoB is a participant of the BoB, which functions as a subchain. It can also be roughly expressed as an "internet of blockchains", where each subchain is as 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 typically are via the mainchain for two-way pegs atomic swaps among the homogeneous blockchains, where all actions should be coordinated by the mainchain. A BoB is more like a notary scheme (in implementation), where the mainchain severs as a notary to record the activities that happen on each subchain, and each subchain can be heterogeneous. There exist several BoB dApp scenarios in the literature. Due to various application scenarios, it is hard to get a general framework on how BoB works, and we discuss the BoB schemes case-by-case.

a) Overledger: Overledger [114] is a kind of blockchain operating system, which allows some general-purpose applications to run on different blockchains. It abstracts a single-ledger dependent technology to overcome different architectures, by introducing a vendor-independent protocol to achieve message-oriented communication. Overledger also allows the business logic to decouple from the underlying ledger, which increases communicability among chains, e.g., with the privacy constraints of dApps.

The architecture of Overledger has four distinct 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, which includes all operations needed to reach an agreement in diverse blockchain domains. Typically, this layer can operate on different ledgers. The messaging layer is a logic layer, which is used to retrieve and store all relevant information from different ledgers. The information communicated in this layer includes transaction data, smart contracts, or metadata (e.g., the digest of out-of-chain messages). It can be considered as 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 and builds messages from transaction information. It provides a filtering service to filter out unnecessary information (e.g., information exchanged in out-of-chain), and orders them into the block. The validation scheme examines the application scenarios and its specification, and those information can be extracted from transaction data. The application layer is the upper part of the reference architecture, interacting with applications, where messages from different applications may be shared or referred to by other applications.

The communication in Overledger is a similar two-phase commit protocol scheme for atomic commitment.

b) HyperService: HyperService [6] is a platform and framework to offer interoperability and programmability across heterogeneous blockchains 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. The 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-blockchain operations which 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, which is 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 the UIP itself contains another two building blocks: *Network Status Blockchain (NSB)* and *Insurance Smart Contracts (ISCs)*. The NSB serves as a blockchain of blockchains to provide an objective and unified view of the dApps' execution status based on the execution of ISCs. While the ISCs revert all executed transactions to guarantee financial atomicity and make misbehaved entities accountable.

c) Hyperledger Fabric: Hyperledger Fabric [104] is a modular and extensible open-source system for running distributed appellations with extensibility. It provides support on modular consensus protocols, and this allows the deployed system to be more customized (e.g., targeting specific use cases and different trust models). The Fabric first introduces a novel *execute-order-validate* blockchain architecture. Practically, a general application deployed on Fabric consists of two main components: a chaincode and an endorsement policy. A chaincode essentially is a smart contract that executes and operates the application logic during execution. The chaincode is the most important component for a distributed applications, while 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, however, untrusted application developers do not have the right to choose or modify those policies. We can roughly consider an endorsement policy as a static library of Fabric for the transaction validation process, which can only be set up (e.g., parameterization) by the chaincode.

Fabric contains modular building blocks for each application: an ordering service, a membership service provider, an optional peer-to-peer gossip service, and smart contracts. A Fabric network enables the feature that multiple distinct blockchains can connect to the same ordering service, in which each blockchain is called a channel. Channels are used to partition the state of the blockchain network, and the order of transactions in each channel is separate. However, different applications can implement different chaincode to facilitate inter-blockchain communication.

d) SMChain: SMChain [115] as a blockchain dApp targets for secure metering applications in distributed industrial plants. It 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 won 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 among 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 the state chain blocks will return to each local chain for integrity and interoperability checks. By allowing a two-layer structure, it can achieve a certain level of interoperability on the status of each local chain. However, the proposed structure is still in prototype and focuses on architecture design, and no real applications are available.

Besides the above-mentioned blockchain of blockchains solutions, there are other works, e.g., Block Collider [116] and CAPER [117]. Block Collider aims to be a multi-chain platform, where transactions

and smart contracts can initiate or exchange by smart contracts on other blockchains. Distributed application developers can modularly combine exotic features from blockchain across the multi-chain platform, and can build in the capability to load-balance work between chains. CAPER is a permissioned blockchain platform to support both internal and cross-blockchain transactions of multiple collaborating dApps, e.g., supply chain applications. Each application-specific blockchain maintains a directed acyclic graph where each application is restricted to only access and maintain its own ledger. It utilizes three specific consensus protocols, to globally maintain and order cross-application transactions, and a transaction may work with different internal consensus protocols in these own ledgers. These three consensus protocols consist of global consensus using a separate set of orders, hierarchical global consensus, and one-level global consensus.

2) Blockchain Adapters: A blockchain adapter more targets end-users (e.g., blockchain clients or blockchain applications), by providing an interface to allow end-users to handle the interoperability, e.g., via runtime selection or smart contracts. There also exist several literature works in this category.

Frauenthaler et al. [118] 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 provides the switch-over service between blockchain even during the runtime. However, it must continuously monitor several blockchains simultaneously. If a more appropriate blockchain (than the current one) comes out, the framework suggests switching to that chain, e.g., by routing subsequent operations to the new blockchain. 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 [119] is another policy-based blockchain selection scheme for interoperability, that follows their previous two policy-based selection work [120] and Bifrost [5]. PleBeuS adopts a generic cost-aware method with consideration on 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 two types, prioritizing the blockchain that is either performance targeted or cost target. This work is a conceptual prototype, and the detailed supported blockchains are not provided.

A *move* protocol based on a smart contract is proposed in the work [121], which enables blockchain interoperability. It offers developers an operational primitive and enables contracts and assets switching between blockchains, with the guarantee of consistency and most key blockchain properties. This move protocol is based on a move operation. The move protocol divides a move operation into two separated transactions, *Move1* and *Move2*. Move1 is used to lock the state of a smart contract in the source blockchain to ensure integrity, while Move2 is used to reconstruct the smart contract on the target blockchain. Similar to two-phase commit protocols, it adds some constraints on the sequence, e.g., only if Move1 transaction has been executed successfully with proof to Move2, the Move2 transaction can proceed successfully.

3) Blockchain Agnostic Protocols: Blockchain agnosticism refers to a single platform allowing multiple chains/blockchains to co-exist, enabling cross-chain or cross-blockchain communication between arbitrarily distributed ledgers. In essence, blockchain agnosticism provides its end users various options to pick their optimal blockchain and provide the capabilities for migrations between blockchains. There also exist several literature works in this direction.

a) Autonomous Systems: A design philosophy for interoperable blockchain ecosystems (analogy to the design philosophy of the Internet infrastructure) is proposed in [40] and an interoperability architecture for blockchain autonomous systems following the design principle is proposed in [40]. Both are blockchain agnostic protocols. The proposed framework is based on autonomous systems (AS) (alternatively called routing domains) as one kind of connectivity unit (like one single participant in blockchain system) to offer the scale-up capability, which also allows routing information to been hierarchically aggregated via intra- or inter-domain routing. 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 inter-connectivity and thus interoperability, in which the gateways are used to execute and validate cross-blockchain transactions. The framework has two kinds of nodes, one is intradomain nodes, which are responsible to maintain ledger information and conduct transactions within one domain, and the other is interdomain nodes (aka interdomain gateways), which is used to handle cross-domain transactions involving different blockchain ASs. Each domain of AS can be owned by a private organization, e.g., in the form of a private blockchain. It is crucial to guarantee the confidentiality of information of each private blockchain, and the framework also provides a use-case example to guarantee this requirement. Both works on AS are still in the prototype stage, and real implementations are still missing.

b) Interledger Protocol: Originally, Interledger protocol (ILP) [85] is designed for a payment network across different payment systems, which provides a way to secure transferring process between ledgers and offers the way to create a direct connection between two ledgers if the ones are with accounts on both ledgers. At the core of ILP is the 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 Byzantine fault-tolerant algorithm to guarantee the consistency of ledger's state.

The new version of ILP, called ILPv4 [122], is an agnostic version, which can be adopted into other distributed ledgers, instead of only on the payment network. ILPv4 is usable across any type of ledgers, even those that were not built for interoperability. A participant in ILPv4 typically 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 ledgers or payment channels for settling bilateral payment obligations, with its packets sending only between connectors, without involving the participation of the underlying ledgers. This means the packets communicated in ILPv4 are based on the 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 typically 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.

Also, there is a Java implementation of Interledger protocol from the Hyperledger project, called Hyperledger *Quilt* [123]. Quilt tries to develop a suite of open protocols and standards that allows payment interoperability across any currency, fiat, or crypto. Until now, Quilt has supported several interledger protocols, e.g., interledger addresses, ILPv4, payment pointers, ILP-over-HTTP, SPSP (Simple Payment Setup Protocol), and STREAM (a protocol reliably sending tokens and exchanging information over ILPv4).

c) Perun: Perun [124] originally is a joint Distributed Ledger Technology (DLT) Layer 2 scaling project, and now joins Hyperledger as a Labs project. Perun is a blockchain-agnostic state channel framework, aiming to make blockchain ready for mass adoption and alleviate current technical challenges, e.g., high fees, latency, and low transaction throughput. Perun 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 the state-channel network intermediaries. Perun enables interoperability via blockchain agnostic design and state-channel virtualization, and this further allows transactions and smart contracts that can be executed across different blockchains.

d) Gravity: Gravity [125] is a blockchain-agnostic cross-chain communication protocol among blockchains and outside entities (e.g., data oracles). The Gravity network consists of a non-isomorphic Gravity node, in which providers can openly choose to operate in one or more target chains, or they can implement extractors to extract necessary data. 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). In general, Gravity can be considered as a singular decentralized blockchain-agnostic oracle.

SuSy [126] is a blockchain-agnostic cross-chain asset transfer gateway protocol based on Gravity, a second layer protocol over Gravity. The current version of SuSy focuses on token transferring without bringing any incentive models for cross-chain transfer providers. Also, a Susy protocol highly relies on the trusted oracle model, which acts as an intermediary in the information transferring process between blockchains. However, both protocols (Gravity and SuSy) are currently in the stage of concept; no implementations are available.

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 [34]. 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 [105], 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.

4) Discussions on 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 blockchain of blockchains 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 records the activities among sub-blockchains. Cross-chain communication 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 the 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 for ones that require a direct connection with a trusted party. Blockchain agnostic solutions are more independent, which offers interoperability to existing blockchains. Most solutions in this category focus on prototype design, with more generalization than the solutions in the blockchain adapter. That means blockchain agnostic protocols provide some flexibility to the adaptation on the selection of blockchains, and the selection does not rely on the underlying blockchains. However, most solutions in this category do not grant backward compatibility.

V. Opportunities

Roughly speaking, blockchain interoperability refers to the ability to share information, operate, and transact across various different blockchain systems, either homogeneous or heterogeneous. In a fully interoperable environment, a participant from one blockchain should have the ability to access and interact with another blockchain with little effort. One general goal to achieve full interoperability requires a system to securely and correctly relay the entities or information between two blockchains in a fully decentralized manner [127]. Many application scenarios will benefit from blockchain interoperability, and blockchain interoperability is regarded as the next major wave to innovate the extension of decentralized Internet. With a decentralized Internet, various blockchains can interconnect with each other to increase the scalability, speed, extensibility, and flexibility of the blockchain technology. This section focuses on the opportunities provided by blockchain interoperability in various applications, and the other technologies that can be promoted by blockchain interoperability.

A. Blockchain Interoperability Applications

Blockchains essentially are cryptographic protocols that allow a network of nodes collectively to maintain a shared ledger of information without the need for complete trust between the nodes. One of their goals is to apply to various practical applications and provide the unique features enhanced by blockchain to these applications. Interoperability provides a way to enable faster, more efficient, and highly secure business-to-business or business-to-consumer transactions across multiple blockchains. Early applications with blockchain have delivered promising results in a broad range of finance and banking industries. In fact, most existing network-based applications can benefit from blockchain interoperability, which aims to freely

exchange information in a trusted, immutable, and decentralized manner. The inseparability of blockchain applications has promoted a new range of smart services that offer significant benefits to its original applications. There are millions of specific applications, and this section lists several typical applications, as examples, that can benefit from blockchain interoperability, e.g., supply chain, healthcare, and industry. Different applications may have different issues to resolve when achieving blockchain interoperability.

The term "supply chain" is a general term, which can be adopted into various applications, e.g., transportation industry, food supply chains, pharmaceutical supply chains, and manufacturing supply chains. To successfully apply blockchain into supply chain applications, several key challenges need to be resolved, namely, traceability, dispute resolution, cargo integrity and security, compliance, and trust and stakeholder management [127]. Traceability allows participants, e.g., 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. Though smart contracts can relieve within an organization, when involving multiple organizations, interoperable smart contracts are not easy to achieve.

One of the area in which blockchain has tremendous influence is healthcare, and blockchain technology has great potential to transform the healthcare ecosystem to a new level. For example, interoperability can enhance clinical care service, by offering the access to some historical clinical data even from other hospitals [128]. The landscape of health interoperability is primarily focused on special organizations like hospitals and clinics, and these internal information infrastructure usually creates and siloes details. The information exchange is pretty rare, e.g., mostly inspired by some financial incentives or research purpose. However, many interoperability issues remain. For example, an interchange between separate organizations can be operationally difficult and requires substantial cooperation among the entities. Data sharing agreement, procedures for patients matching should be agreed upon before actual data can be exchanged. Besides, there are numerous technical barriers, e.g., authentication and privacy-preserving schemes [129]. The issues and challenges can be enhanced with interoperable blockchain systems.

Industrial processes typically require multiple entity collaborations, and each entity can build its own blockchain system. Without information sharing, each blockchain system would be like an isolated island, and the potential collaboration will be 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, e.g., platform, data confidentiality, data privacy, and application-specificity. The current industrial blockchain platforms lack a design standard to interoperate the blockchains and a collaborative environment.

Again, there exist many other blockchain applications that require interoperability. Each of these applications may have some special requirements. When designing an interoperable blockchain system, these specific requirements should be considered during the prototype design.

B. Decentralized Blockchain Internet

Following design principles of the Internet, one of the ultimate goals of blockchain interoperability is to create a decentralized Internet, in which each blockchain application can facilitate the packet switch communication without considering the underlying infrastructure of different blockchain systems. As pointed out in [130], the Internet should have several fundamental goals to achieve, namely, survivability, a variety of service types, and a variety of networks. Survivability ensures that connectivity across the Internet should not be compromised even under some network failures, e.g., loss of some gateways. A variety of service types means that the Internet must support multiple communications services. And, a variety of networks indicates that the Internet must accommodate a variety of networks.

The decentralized Internet distributes the control of Internet to the users, which provides the fair chance of participation or distribution of network resources. Constructing a decentralized Internet requires different kinds of components in the perspective of blockchain's promise of decentralization and distributed trust, namely, decentralized naming and discovery systems, routing in the decentralized Internet, and decentralized storage [131] [132]. Blockchain itself offers a namespace system, in which users can append the transactions to this ledger system with some unique guarantees, e.g., tamper-resistance, immutability, availability, and transparency. However, some challenges still remain to provide a secure and distributed naming service, such as key management [133]. The interoperability of many distinct (that is, largely isolated and self-contained) blockchain networks will pose a problem if they come together to enable a blockchain-powered decentralized web. The routing mechanism should have to take care of various blockchain features and has the ability to route a transaction between 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 blockchain networks. Decentralized storage requires that the users can securely and privately store their data without disclosing it to any untrusted entities. There exist several decentralized storage solutions, e.g., Storj [134], Inter-Planetary File System (IPFS) [135]. The challenging task of applying such situations to the large-scale decentralized Internet is known as the issue of 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 decentralized Internet.

C. Standardization

There are currently no standards for establishing compatible architectures for blockchain interoperability. Without the available standardization to regulate distinct blockchains, it is difficult or impossible to achieve a service agreement and thus an interoperable system on the integrated processes of blockchains. Moreover, each organization may develop incompatible standards among these partners. This further blocks the progress of blockchain interoperability.

The current progress on blockchain interoperability is still in the early stage, and there is as yet simply no agreement as to which features as de facto blockchain interoperability, nor is there broad agreement on a reference architecture [136]. Without it, independent developments would highly impact achieving an interoperable

blockchain system in the future. 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 developments and implementations may have similar or overlapping functionalities by different organizations (or vendors), it is highly recommended to have some international organizations to control these processes.

Interoperability is only in its early stages, and many research efforts needs to be done. It is no doubt that a single party cannot have the ability to resolve all the issues of blockchain interoperability and to coordinate the attempts of industry organizations and academic researchers to specify viable commercial solutions. Although many promising examples of interoperability across multiple blockchain systems are being achieved, most of these solutions are being carried out on and with centralized databases instead of decentralized ones [137].

VI. Challenges to Blockchain Interoperability

This section explores the challenges of achieving an interoperable blockchain ecosystem. Instead of discussion from the perspectives of technical details, we pose some critical challenges from a high-level perspective.

A. Survivability

As stated in [40], survivability is the key to the success of blockchain interoperability. Survivability means that the transactions from an end-user application (e.g., by a smart contract or an original blockchain) to 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 communication application, and should be within a reasonable delay. In blockchain cases, the features of reliability and the "best-effort delivery" are challenging to maintain, and it is hard to guarantee the application-level transaction can be completed within a reasonable time (possibly independent of an actual blockchain deployment). The communications over multiple blockchain networks should be "connectionless" which means, from a high-level perspective, one blockchain does not need to care about if that transaction has been executed or not in another blockchain. However, there must have some underlying mechanism to guarantee consistency among blockchains, and these underlying mechanisms should be transparent to its user "blockchain".

One of the obstacles of application-level survivability is that a transaction may get executed and confirmed on some blockchains, however, this transaction should be get confirmed independently on all related blockchains, and the application should be kept transparent for this process. This may cause an inconsistent state among multiple blockchains systems. To achieve survivability, it also needs to handle many issues, e.g., reliability, semantic types of blockchain, distinguishability of blockchain systems. The reliability for interoperable blockchain systems 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 application layer, blockchain network layer, or even some hidden middle layer). A semantic type of blockchain 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 speed and achieved the majority of confirmation of a given system, the directionality of transactions, and strength of consensus. Also, there exist a variety of blockchain systems and they may have different supporting infrastructures, computational resources, etc. Thus, achieving Internet-like blockchain interoperability still has a long way to go.

B. Trustless Technology

Blockchain is commonly considered as a reliable and confident machine, which is based on "trustless trust". This feature typically can be achieved by the technique of deterministic execution [138]. A blockchain system itself can guarantee trustworthiness among the participants via various protocols, such as a consensus protocol, to ensure the operations of the overall system. Even though there is no centralized trusted authority, 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 the communication, an intermediary "trusted" (or "virtual trusted") entity is required among distinct blockchain systems.

Even many existing interoperability schemes remove the use of a single centralized trust entity, and instead use a distributed trust, however, this kind of effort is still considered to be a kind of trusted scheme. In a decentralized trust, trust is disseminated to a decentralized network, and no entity has the sole power of monopoly over the act of transacting. The absence of a trusted entity in charge of managing and coordinating interactions over multiple blockchain networks does not, in and of itself, make it a "trustless technology". In these designs, the trust is still not completely removed, but is shifted from intermediaries to technology (e.g., peer-to-peer network, smart contracts, etc) to guarantee a sufficient level of confidence in any blockchain-based applications operating on top of that network [139]. Essentially, a consensus over multiple blockchains is still required. To achieve fully decentralized communication, and thus to fully eliminate the trust, among multiple blockchain systems, still has a long way to go.

C. From Theory to Practice

As discussed earlier, many blockchain application scenarios require interoperability with the use cases from finance to industry, and to economics. Most existing blockchain interoperability solutions are still in theory (or some with prototype demonstration), and few have a real implementation. One reason for this is that the theoretical advances on blockchain interoperability have still not been agreed upon, and each organization may develop and deploy interoperable blockchain solutions based on their 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 [140]. A practical implementation needs to consider and evaluate different evaluation metrics, e.g., throughput, latency, scalability, cost, security, and privacy, and would help in speeding up

the development process and the overall advancement of blockchain interoperability.

Many variants, such as consensus algorithm, computation and communication capabilities of consensus participants, or even peer-to-peer network delay, may affect a correct cross-chain operation. Especially if timelocks are used, assets may be locked forever. Many proposed protocols are still in the theoretical or prototype stage, while 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 these aspects affects the progress of blockchain interoperability. Thus, developers of blockchain interoperability platforms should conduct empirical studies and establish benchmarking data about the platform being developed.

D. Attacks Mitigation Technology

Typically, each independent blockchain has a well-defined security model of its own, and the security model that worked 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 the adversary is less than 33%. Due to the different criteria to evaluate the ability of an adversary, the ability to accumulate state may be lower than that of the accumulation computational power, or vice-versa [141]. If considering the permissionless public blockchain, which may not be Sybil resistant [142], this would make the interoperability among blockchains more difficult [48]. In a cross-blockchain setting, it is almost impossible to detect and countermeasure the bribing attacks executed cross-chain [143].

Any blockchain should prevent replay attacks, where a transaction or request is re-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 and countermeasure are difficult [144]. For example, in a single blockchain, protection can involve the use of a sequence number or each participant keeps track of previously processed proofs of transaction. There is no such mechanism to detect this attack in multiple blockchains [145]. Besides, multiple blockchain systems must carefully take care of composability attacks, which are related to the stability of a consensus ((e.g., with the probability of a reversion being negligible)) [146]. For example, how many numbers of blockchain or confirmation should a transaction have before being accepted as secure [140]. In multiple blockchain systems, it may be insufficient to consider the composition of a block within a single blockchain, which also requires consideration of the state update in the other blockchains.

Besides the above-mentioned challenges, there exist many other challenges, e.g., compatible cryptographic primitives and collateralization. Different blockchains may leverage different cryptographic schemes or even different scenarios of the same scheme, and compatible and reliable 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 [147] [148]. It is crucial and difficult to guarantee that the available collateral has sufficient value to outweigh potential gains from misbehavior [48].

Blockchain interoperability is still in its infancy, and thus for from its practical implementation, which certainly 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 implementation efforts are needed to enhance the standardization of blockchain interoperability.

VII. Potential Research

This section provides the potential research directions to improve the flexibility of blockchain interoperability, and further standardize these approaches.

A. Interoperable Architecture

Nowadays, given that many blockchain interoperability solutions and platforms are available, these solutions are still separated for specific purposes, and no standardized interoperable architecture has been provided. Until now, there is no formal definition of blockchain interoperability, nor of an interoperable architecture. To promote the development of a decentralized Internet by utilizing blockchains, it is necessary to first model the interoperability at various layers (e.g., following the Internet's OSI model [149]). A modeling process must consider the variety of applications, since different applications may have different 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 high degree of heterogeneity and diversity of various applications and services, it is a challenge to achieve an interoperable architecture to support validating data and processes. For example, different applications may have different factors that affect overall interoperability, such as differences in consensus protocols, block sizes, and interval of generating blocks [150]. To achieving an interoperable blockchain architecture, the designers, at least, need to consider several aspects, e.g., distribution, processing of big data, heterogeneity, dynamicity, and mobility. Furthermore, since the participants of interoperable blockchains over a decentralized Internet would be geographically distributed, the amount of data to be processed would be increased exponentially with an increasing number of participants. The systems of the participants may be heterogeneous, and the services a participant accesses may be very dynamic and mobile. All these factors affect the achievement of an interoperable architecture.

B. Cross-blockchain Primitives

Cross-blockchain interaction remains still an active research direction, and most approaches still continue to rely on atomic swaps. More general approaches on cross-blockchain data or information exchange, including atomic tokens/assets transfer, would help to dissolve the current fragmentation of research. We categorize current solutions of blockchain interoperability issues into three categories: chain-based interoperability, bridge-based interoperability, and dApp-based interoperability. Each solution can be considered as an independent model to meet partial requirements of blockchain interoperability. Still, there may exist other cross-blockchain primitives waiting 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 some drawbacks in some specific solutions. These may need to explore the integration of other architecture to optimize the current structure. 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 own cross-chain communication scheme [151].

The success of blockchain interoperabilities highly depends on the process of cross-blockchain communication, and hence the cross-blockchain primitives. A good cross-blockchain primitive should have several features, e.g., eliminating a centralized entity, enabling multi-party transactions, reasonable performance, and portable interface. Further, the cross-blockchain 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 [152]. Most existing cross-blockchain schemes still focus on two-party scenarios, and a multi-party transaction and communication scheme, such as a multi-party atomic cross-chain swap, is still missing. Considering the performance of cross-blockchain communication, an acceptable response time will also be required. The cross-blockchain primitives also are portable. For example, when the participants concurrently work with multiple cross-blockchain platforms, portability issues may be created for different kinds of interfaces. Thus, a common interface with different blockchains and their participants is preferable. Besides, a good cross-blockchain primitive should be provable, secure, correct, and atomic, which requires strict proofs to theoretically show it works.

C. Security and Privacy

When multiple blockchains work together, security and privacy are necessary considerations. In general, different blockchains may adopt different security primitives, in which one security primitive is secure in one system, but is not secure in another system. Security is still the major concern for the willingness to adopt interoperability among stakeholders. It is highly recommended to develop security standards for scripting smart contracts and other blockchain primitives. Also, the privacy-preserving technologies in current blockchain systems are not robust enough. The ideal solution for preserving privacy in multiple blockchain systems would be in a form of decentralized record-keeping that is completely obfuscated and anonymous by design. Such solutions may need to consider different choices of blockchains, such as public blockchains or private blockchains.

Privacy is crucial in any sensitive interaction (e.g., financial assets or health records) and thus in cross-chain communication. Ideally, it should not be possible for an observer to determine what activities or events have been synchronized across blockchains. However, 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 initial a cross-chain transaction to a secure blockchain system to acquire sensitive data or even launch an attack on the secure blockchain system. Thus, security and privacy-assisted technologies are required to succeed in blockchain interoperability.

D. Scalability

Scalability is not only an issue of blockchain interoperability, but also an issue of blockchain itself. The key features of the blockchain (e.g., decentralization and immutability) require that every full node store a full copy of the blockchain; however, this comes at a cost of scalability. The scalability issue in blockchain 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 number of concurrent workloads [153] [154]. In the current design, many blockchain systems are still suffering from poor throughput. Scaling blockchain has become an active research area [155], for example, via increased block size [156] and sharding techniques [151]. Blockchain scalability issues are still an open research area, and many different initiatives and efforts in recent research are aimed at improving blockchain scalability, from layered chain structure to sharding techniques [151].

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 [12]. 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 IV-A1. The child-chain solution has a parent-child structure, processes the transactions in the child-chain, and records the results in the parent-chain, as the structure of SMChain, described in Section IV-C. The Inter-chain solution provides a way to enable communication among the various blockchains, either homogeneous or heterogeneous, whose infrastructure is like the side-chain solution. Except for the on-chain solution, which only handles one blockchain, other solutions can help the design of interoperable blockchain infrastructures.

Besides the above-mentioned research directions on blockchain interoperability, there are other hot research areas, e.g., standardization, usability, and reachability.

VIII. Conclusion

The research and progress on blockchain interoperability are still in their infancy stage. This paper presents a Systematization of Knowledge for the existing efforts on blockchains interoperability. We classify them into several key categories, namely, chain-based interoperability, bridge-based interoperability, and dApp-based interoperability. For each category, we review and study the state-of-the-art solutions with detailed analysis, e.g., on advantages and disadvantages. This paper serves as a starting point for exploring blockchain interoperability. Based on what we observed and learned, we discussed opportunities and challenges when applying blockchain interoperability into current blockchain design. Finally, we provide several potential research directions that can help to advance an interoperable blockchain ecosystem.

References

[1] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[3] W. Nikolakis, L. John, and H. Krishnan, "How blockchain can shape sustainable global value chains: an evidence, verifiability, and enforceability (eve) framework," *Sustainability*, vol. 10, no. 11, p. 3926, 2018.

[4] P. Wegner, "Interoperability," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 285–287, 1996.

[5] E. J. Scheid, T. Hegnauer, B. Rodrigues, and B. Stiller, "Bifröst: a modular blockchain interoperability api," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 332–339.

[6] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 549–566.

[7] M. Borkowski, C. Ritzer, D. McDonald, and S. Schulte, "Caught in chains: claim-first transactions for cross-blockchain asset transfers," *Technische Universität Wien, Whitepaper*, 2018.

[8] S. Schulte, M. Sigwart, P. Frauenthaler, and M. Borkowski, "Towards blockchain interoperability," in *International Conference on Business Process Management*. Springer, 2019, pp. 3–10.

[9] M. Herlihy, "Atomic cross-chain swaps," in *Proceedings of the 2018 ACM symposium on principles of distributed computing*, 2018, pp. 245–254.

[10] M. H. Miraz and D. C. Donald, "Atomic cross-chain swaps: development, trajectory and potential of non-monetary digital token swap facilities," *Annals of Emerging Technologies in Computing (AETiC) Vol*, vol. 3, 2019.

[11] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE transactions on industrial informatics*, vol. 14, no. 8, pp. 3690–3700, 2017.

[12] S. Kim, Y. Kwon, and S. Cho, "A survey of scalability solutions on blockchain," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 1204–1207.

[13] V. Buterin, "Chain interoperability," *R3 Research Paper*, 2016.

[14] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *arXiv preprint arXiv:2005.14282*, 2020.

[15] X. Shen, H. Yu, J. Buford, and M. Akon, *Handbook of peer-to-peer networking*. Springer Science & Business Media, 2010, vol. 34.

[16] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[17] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT professional*, vol. 19, no. 4, pp. 68–72, 2017.

[18] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2018.

[19] R. Lai and D. L. K. Chuen, "Blockchain–from public to private," in *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*. Elsevier, 2018, pp. 145–177.

[20] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for iot," *IEEE Access*, vol. 7, pp. 38 431–38 441, 2019.

[21] T. Härder, "Dbms architecture–still an open problem," *Datenbanksysteme in Business, Technologie und Web, 11. Fachtagung des GIFachbereichs "Datenbanken und Informationssysteme"(DBIS)*, 2005.

[22] S. Tai, J. Eberhardt, and M. Klems, "Not acid, not base, but salt," in *Proceedings of the 7th International Conference on Cloud Computing and Services Science*. SCITEPRESS-Science and Technology Publications, Lda, 2017, pp. 755–764.

[23] D. Zhao and T. Li, "Distributed cross-blockchain transactions," *arXiv preprint arXiv:2002.11771*, 2020.

[24] D. Pritchett, "Base: An acid alternative," *Queue*, vol. 6, no. 3, pp. 48–55, 2008.

[25] M. Kühne, "Extending cross-blockchain token transfers," Ph.D. dissertation, Wien, 2020.

[26] C. Xie, C. Su, M. Kapritsos, Y. Wang, N. Yaghmazadeh, L. Alvisi, and P. Mahajan, "Salt: Combining {ACID} and {BASE} in a distributed database," in *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, 2014, pp. 495–509.

[27] R. Han, H. Lin, and J. Yu, "On the optionality and fairness of atomic swaps," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 62–75.

[28] T. Hegnauer, "Design and development of a blockchain interoperability api," Ph.D. dissertation, Master's thesis, CSG@ IFI, University of Zurich, Switzerland, to appear 2019 . . . , 2019.

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

[30] I. Tsabary, M. Yechieli, and I. Eyal, "Mad-htlc: because htlc is crazy-cheap to attack," *arXiv preprint arXiv:2006.12031*, 2020.

[31] L. Lys, A. Micoulet, and M. Potop-Butucaru, "Atomic cross chain swaps via relays and adapters," in *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, 2020, pp. 59–64.

[32] J.-Y. Zie, J.-C. Deneuville, J. Briffaut, and B. Nguyen, "Extending atomic cross-chain swaps," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2019, pp. 219–229.

[33] V. Zakhary, D. Agrawal, and A. E. Abbadi, "Atomic commitment across blockchains," *arXiv preprint arXiv:1905.02847*, 2019.

[34] H. Wang, Y. Cen, and X. Li, "Blockchain router: a cross-chain communication protocol," in *Proceedings of the 6th international conference on informatics, environment, energy and applications*, 2017, pp. 94–97.

[35] P. Robinson, "Consensus for crosschain communications," *arXiv preprint arXiv:2004.09494*, 2020.

[36] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[37] A. Garoffolo, D. Kaidalov, and R. Oliynykov, "Zendoo: a zk-snark verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains," *arXiv preprint arXiv:2002.01847*, 2020.

[38] A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," in *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016, pp. 281–282.

[39] M. Borkowski, D. McDonald, C. Ritzer, and S. Schulte, "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*, 2018.

[40] T. Hardjono, A. Lipton, and A. Pentland, "Toward an interoperability architecture for blockchain autonomous systems," *IEEE Transactions on Engineering Management*, 2019.

[41] D. Bradbury, "The problem with bitcoin," *Computer Fraud & Security*, vol. 2013, no. 11, pp. 5–8, 2013.

[42] M. Nissl, E. Sallinger, S. Schulte, and M. Borkowski, "Towards cross-blockchain smart contracts," *arXiv preprint arXiv:2010.07352*, 2020.

[43] M. Borkowski, M. Sigwart, P. Frauenthaler, T. Hukkinen, and S. Schulte, "Dextt: Deterministic cross-blockchain token transfers," *IEEE Access*, vol. 7, pp. 111 030–111 042, 2019.

[44] M. Sigwart, P. Frauenthaler, C. Spanring, and S. Schulte, "Towards cross-blockchain smart contracts."

[45] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1203–1211.

[46] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White Paper*, 2016.

[47] J. Kwon and E. Buchman, "Cosmos: A network of distributed ledgers," *URL https://cosmos. network/whitepaper*, 2016.

[48] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, "Sok: communication across distributed ledgers." 2019.

[49] H. Pagnia and F. C. Gärtner, "On the impossibility of fair exchange without a trusted third party," Technical Report TUD-BS-1999-02, Darmstadt University of Technology ..., Tech. Rep., 1999.

[50] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the 1997 workshop on New security paradigms*, 1998, pp. 48–60.

[51] D. Skeen, "Nonblocking commit protocols," in *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, 1981, pp. 133–142.

[52] F. Cristian, "Synchronous and asynchronous," *Communications of the ACM*, vol. 39, no. 4, pp. 88–97, 1996.

[53] O. Shalev and N. Shavit, "Split-ordered lists: Lock-free extensible hash tables," *Journal of the ACM (JACM)*, vol. 53, no. 3, pp. 379–405, 2006.

[54] C. Egger, P. Moreno-Sanchez, and M. Maffei, "Atomic multi-channel updates with constant collateral in bitcoin-compatible payment-channel networks," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 801–815.

[55] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," 1996.

[56] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *Annual international cryptology conference*. Springer, 2018, pp. 757–788.

[57] M. Khabbazian, T. Nadahalli, and R. Wattenhofer, "Outpost: A responsive lightweight watchtower," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 31–40.

[58] R. M. Parizi, S. Homayoun, A. Yazdinejad, A. Dehghantanha, and K.-K. R. Choo, "Integrating privacy enhancing techniques into blockchains using sidechains," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 2019, pp. 1–4.

[59] L. Deng, H. Chen, J. Zeng, and L.-J. Zhang, "Research on cross-chain technology based on sidechain and hash-locking," in *International Conference on Edge Computing*. Springer, 2018, pp. 144–151.

[60] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *Journal of Network and Computer Applications*, vol. 149, p. 102471, 2020.

[61] J. Teutsch, M. Straka, and D. Boneh, "Retrofitting a two-way peg between blockchains," *arXiv preprint arXiv:1908.03999*, 2019.

[62] K. Ohta and T. Okamoto, "Multi-signature schemes secure against active insider attacks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 82, no. 1, pp. 21–31, 1999.

[63] J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, and M. Friedenbach, "Strong federations: An interoperable blockchain solution to centralized third-party risks," *arXiv preprint arXiv:1612.05491*, 2016.

[64] A. Kiayias, N. Lamprou, and A.-P. Stouka, "Proofs of proofs of work with sublinear complexity," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 61–78.

[65] M. Szydlo, "Merkle tree traversal in log space and time," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2004, pp. 541–554.

[66] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, p. 72, 2014.

[67] Loom, "Intro to loom network — loom sdk," https://loomx.io/developers/en/intro-to-loom.html, 2016.

[68] S. D. Lerner, "Rsk white paper overview," 2015.

[69] J. Nick, A. Poelstra, and G. Sanders, "Liquid: A strongly federated asset issuance platform," 2019.

[70] P. Network, "Poa-network-whitepaper," *Accessed*, vol. 10, no. 3, p. 19, 2018.

[71] Binance, "Binance chain (dex)," *Version 1.1, accessed Dec., 2020 URL https://docs.binance.org/*, 2019.

[72] D. Larimer, "Delegated proof-of-stake (dpos)," *Bitshare whitepaper*, 2014.

[73] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[74] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, 2016.

[75] S. D. Lerner, "Drivechains, sidechains and hybrid 2-way peg designs," 2016.

[76] F. Vogelsteller and V. Buterin, "Eip 20: Erc-20 token standard," 2015.

[77] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, pp. 1–47, 2017.

[78] A. Culwick and D. Metcalf, "The blocknet design specification," 2018.

[79] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[80] D. D. F. Maesa and P. Mori, "Blockchain 3.0 applications survey," *Journal of Parallel and Distributed Computing*, vol. 138, pp. 99–114, 2020.

[81] K. Wang, Z. Zhang, and H. S. Kim, "Reviewchain: Smart contract based review system with multi-blockchain gateway," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1521–1526.

[82] K. Qin and A. Gervais, "An overview of blockchain scalability, interoperability and sustainability," *Hochschule Luzern Imperial College London Liquidity Network*, 2018.

[83] C. Cachin and M. Vukolić, "Blockchain consensus protocols in the wild," *arXiv preprint arXiv:1707.01873*, 2017.

[84] D. Balazs, "Herdius whitepaper," *Version 1.1, accessed Dec., 2020 URL https://herdius.com/whitepaper/Herdius*$_{Technical}$*$_{Paper.pdf}$*, 2017.

[85] S. Thomas and E. Schwartz, "A protocol for interledger payments," *URL https://interledger. org/interledger. pdf*, 2015.

[86] M. Spoke, N. Team *et al.*, "Aion: Enabling the decentralized internet," *AION, White Paper, Jul*, 2017.

[87] B. Pillai, K. Biswas, and V. Muthukkumarasamy, "Blockchain interoperable digital objects," in *International Conference on Blockchain*. Springer, 2019, pp. 80–94.

[88] T. Nolan, "Alt chains and atomic transfers," in *Bitcoin Forum, May*, 2013.

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

[90] M. Herlihy, B. Liskov, and L. Shrira, "Cross-chain deals and adversarial commerce," *arXiv preprint arXiv:1905.09743*, 2019.

[91] B. Yu, S. K. Kermanshahi, A. Sakzad, and S. Nepal, "Chameleon hash time-lock contract for privacy preserving payment channel networks," in *International Conference on Provable Security*. Springer, 2019, pp. 303–318.

[92] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 455–471.

[93] M. H. Ameri, M. Delavar, J. Mohajeri, and M. Salmasizadeh, "A key-policy attribute-based temporary keyword search scheme for secure cloud storage," *IEEE Transactions on Cloud Computing*, 2018.

[94] B. Dai, S. Jiang, M. Zhu, M. Lu, D. Li, and C. Li, "Research and implementation of cross-chain transaction model based on improved hash-locking," in *International Conference on Blockchain and Trustworthy Systems*. Springer, 2020, pp. 218–230.

[95] Y. Zhang, D. Yang, and G. Xue, "Cheapay: An optimal algorithm for fee minimization in blockchain-based payment channel networks," in

ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019, pp. 1–6.

[96] COMIT, "Comit protocol stack," *URL https://comit.network/docs/comit-protocol/comit-protocol-stack/*, 2020.

[97] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability." in *NDSS*, 2019.

[98] A. Miller, I. Bentov, R. Kumaresan, and P. McCorry, "Sprites: Payment channels that go faster than lightning," *CoRR abs/1702.05812*, vol. 306, 2017.

[99] M. Black, T. Liu, and T. Cai, "Atomic loans: Cryptocurrency debt instruments," *arXiv preprint arXiv:1901.05117*, 2019.

[100] P. Frauenthaler, M. Sigwart, C. Spanring, and S. Schulte, "Testimonium: A cost-efficient blockchain relay," *arXiv preprint arXiv:2002.12837*, 2020.

[101] J. Chow, "Btc relay," *btc-relay*, 2016.

[102] L. Luu, N. Rush, and N. Lin, "Peacerelay: Connecting the many ethereum blockchains," *Retrieved Oct*, vol. 15, p. 2019, 2019.

[103] Hyperledger, "Hyperledger cactus whitepaper," *Accessed Dec, 2020 URL https://github.com/hyperledger/cactus*, 2020.

[104] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[105] L. Kan, Y. Wei, A. H. Muhammad, W. Siyuan, G. Linchao, and H. Kai, "A multiple blockchains architecture on inter-blockchain communication," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2018, pp. 139–145.

[106] G. Falazi, U. Breitenbücher, F. Daniel, A. Lamparelli, F. Leymann, and V. Yussupov, "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, 2020, pp. 134–149.

[107] I. Bentov, Y. Ji, F. Zhang, L. Breidenbach, P. Daian, and A. Juels, "Tesseract: Real-time cryptocurrency exchange using trusted hardware," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1521–1538.

[108] J. Burdges, A. Cevallos, P. Czaban, R. Habermeier, S. Hosseini, F. Lama, H. K. Alper, X. Luo, F. Shirazi, A. Stewart *et al.*, "Overview of polkadot and its design considerations," *arXiv preprint arXiv:2005.13456*, 2020.

[109] J. Kwon and E. Buchman, "Cosmos whitepaper," 2019.

[110] Wanchain, "Wanchain: Building super financial markets for the new digital economy," *Whitepaper Version 0.9.1 URL https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf*, 2017.

[111] ARK, "Ark ecosystem whitepaper," *Version 2.1.0 URL https://ark.io/Whitepaper.pdf*, 2019.

[112] A. Stewart and E. Kokoris-Kogia, "Grandpa: a byzantine finality gadget," *arXiv preprint arXiv:2007.01560*, 2020.

[113] Wanchain, "Wanchain 4.0 t-bridge framework," *URL https://www.wanchain.org/learn/*, 2020.

[114] G. Verdian, P. Tasca, C. Paterson, and G. Mondelli, "Quant overledger whitepaper," 2018.

[115] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Smchain: A scalable blockchain protocol for secure metering systems in distributed industrial plants," in *Proceedings of the International Conference on Internet of Things Design and Implementation*. ACM, 2019, pp. 249–254.

[116] B. C. Team, "Block collier whitepaper," *Version 0.9.9 URL https://overline.network/blockcollider*$_i$*itepaper.pdf*, 2018.

[117] M. J. Amiri, D. Agrawal, and A. E. Abbadi, "Caper: a cross-application permissioned blockchain," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1385–1398, 2019.

[118] P. Frauenthaler, M. Borkowski, and S. Schulte, "A framework for assessing and selecting blockchains at runtime," in *2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. IEEE, 2020, pp. 106–113.

[119] E. J. Scheid, D. Lakic, B. B. Rodrigues, and B. Stiller, "Plebeus: a policy-based blockchain selection framework," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–8.

[120] E. Scheid, B. Rodrigues, and B. Stiller, "Toward a policy-based blockchain agnostic framework," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 609–613.

[121] E. Fynn, A. Bessani, and F. Pedone, "Smart contracts on the move," *arXiv preprint arXiv:2004.05933*, 2020.

[122] Interledger, "Interledger protocol v4," *Accessed Dec., 2020 URL https://interledger.org/rfcs/0027-interledger-protocol-4/*, 2020.

[123] Hyperledger, "Hyperledger quilt," *Accessed Dec., 2020 URL https://wiki.hyperledger.org/display/quilt/Hyperledger+Quilt*, 2020.

[124] P. Team, "Hyperledger labs," *Accessed Dec., 2020 URL https://www.hyperledger.org/category/hyperledger-labs*, 2020.

[125] A. Pupyshev, D. Gubanov, E. Dzhafarov, I. Kardanov, V. Zhuravlev, S. Khalilov, M. Jansen, S. Laureyssens, I. Pavlov, S. Ivanov *et al.*, "Gravity: a blockchain-agnostic cross-chain communication and data oracles protocol," *arXiv preprint arXiv:2007.00966*, 2020.

[126] A. Pupyshev, E. Dzhafarov, I. Sapranidi, I. Kardanov, S. Khalilov, and S. Laureyssens, "Susy: a blockchain-agnostic cross-chain asset transfer gateway protocol based on gravity," *arXiv preprint arXiv:2008.13515*, 2020.

[127] Y. Chang, E. Iakovou, and W. Shi, "Blockchain in global supply chains and cross border trade: a critical synthesis of the state-of-the-art, challenges and opportunities," *International Journal of Production Research*, vol. 58, no. 7, pp. 2082–2099, 2020.

[128] B. P. Center, "Clinician perspectives on electronic health information sharing for transitions of care," *Washington, DC: Bipartisan Policy Center*, 2012.

[129] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability," *Computational and structural biotechnology journal*, vol. 16, pp. 224–230, 2018.

[130] T. Hardjono, A. Lipton, and A. Pentland, "Towards a design philosophy for interoperable blockchain systems," *arXiv preprint arXiv:1805.05934*, 2018.

[131] M. Ali, R. Shea, J. Nelson, and M. J. Freedman, "Blockstack: A new decentralized internet," *Whitepaper, May*, 2017.

[132] A. Ali, S. Latif, J. Qadir, S. Kanhere, J. Singh, J. Crowcroft *et al.*, "Blockchain and the future of the internet: A comprehensive review," *arXiv preprint arXiv:1904.00733*, 2019.

[133] S. Angieri, A. García-Martínez, B. Liu, Z. Yan, C. Wang, and M. Bagnulo, "A distributed autonomous organization for internet address management," *IEEE Transactions on Engineering Management*, 2019.

[134] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.

[135] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[136] D. Hyland-Wood and S. Khatchadourian, "A future history of international blockchain standards," *The Journal of the British Blockchain Association*, vol. 1, no. 1, p. 3724, 2018.

[137] J. Flood and A. McCullagh, "Blockchain's future: can the decentralized blockchain community succeed in creating standards?" *The Knowledge Engineering Review*, vol. 35, 2020.

[138] P. De Filippi, M. Mannan, and W. Reijers, "Blockchain as a confidence machine: The problem of trust & challenges of governance," *Technology in Society*, vol. 62, p. 101284, 2020.

[139] A. Shahaab, R. Maude, C. Hewage, and I. Khan, "Blockchain-a panacea for trust challenges in public services? a socio-technical perspective," *The Journal of The British Blockchain Association*, p. 14128, 2020.

[140] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," in *Annual International Cryptology Conference*. Springer, 2017, pp. 291–323.

[141] J. Bonneau, E. W. Felten, S. Goldfeder, J. A. Kroll, and A. Narayanan, "Why buy when you can rent? bribery attacks on bitcoin consensus," 2016.

[142] T. Duong, L. Fan, and H.-S. Zhou, "2-hop blockchain: Combining proof-of-work and proof-of-stake securely," *Cryptology ePrint Archive, Report 2016/716*, 2016.

[143] P. McCorry, A. Hicks, and S. Meiklejohn, "Smart contracts for bribing miners," in *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 3–18.

[144] A. Sonnino, S. Bano, M. Al-Bassam, and G. Danezis, "Replay attacks and defenses against cross-shard consensus in sharded distributed ledgers," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 294–308.

[145] P. McCorry, E. Heilman, and A. Miller, "Atomically trading with roger: Gambling on the success of a hardfork," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 334–353.

[146] M. Vukolic, "Eventually returning to strong consistency." *IEEE Data Eng. Bull.*, vol. 39, no. 1, pp. 39–44, 2016.

[147] A. Rosales, "Radical rentierism: gold mining, cryptocurrency and commodity collateralization in venezuela," *Review of International Political Economy*, vol. 26, no. 6, pp. 1311–1332, 2019.

[148] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt, "Xclaim: Trustless, interoperable, cryptocurrency-backed assets," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 193–210.

[149] Y. Yemini, "The osi network management model," *IEEE Communications Magazine*, vol. 31, no. 5, pp. 20–29, 1993.

[150] W. Viriyasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "New blockchain-based architecture for service interoperations in internet of things," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 4, pp. 739–748, 2019.

[151] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Sok: Sharding on blockchain," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. ACM, 2019, pp. 41–61.

[152] X. Wang, O. T. Tawose, F. Yan, and D. Zhao, "Distributed nonblocking commit protocols for many-party cross-blockchain transactions," *arXiv preprint arXiv:2001.01174*, 2020.

[153] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.

[154] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[155] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125.

[156] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 3–16.