Siris, Vasilios A.; Nikander, Pekka; Voulgaris, Spyros; Fotiou, Nikos; Lagutin, Dmitrij; Polyzos, George C.

Interledger Approaches

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2019.2926880

Published: 01/01/2019

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Siris, V. A., Nikander, P., Voulgaris, S., Fotiou, N., Lagutin, D., & Polyzos, G. C. (2019). Interledger Approaches. *IEEE Access*, *7*, 89948-89966. Article 8755830. https://doi.org/10.1109/ACCESS.2019.2926880

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 10, 2019, accepted June 10, 2019, date of publication July 4, 2019, date of current version July 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926880

Interledger Approaches

VASILIOS A. SIRIS[1], (Member, IEEE), PEKKA NIKANDER[2], SPYROS VOULGARIS[1], NIKOS FOTIOU[1], DMITRIJ LAGUTIN[2], AND GEORGE C. POLYZOS[1], (Member, IEEE)

[1]Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, 104 34 Athens, Greece
[2]Department of Communications and Networking, School of Electrical Engineering, Aalto University, 02150 Espoo, Finland

Corresponding author: Vasilios A. Siris (vsiris@aueb.gr)

This work was supported by the project Secure Open Federation for Internet Everywhere (SOFIE), funded by the EU's Horizon 2020 Programme under Grant 779984.

ABSTRACT While blockchains and more generally distributed ledger technologies (DLTs) are passing over their hype curve peak, their shortcomings are becoming more apparent. One relatively recent approach to address their performance, scalability, privacy, and other problems are to use multiple different DLTs instead of relying on just one. While there are no really established standards for combining several DLTs, a few repeating patterns can be observed. In this paper, we present a survey of interledger approaches, discussing and comparing their underlying mechanisms. A shared motivation for all of the discussed interledger solutions is to move away from the "one chain rules them all" model to one that allows the interconnection of multiple ledgers, with different features and advantages, while also supporting innovation. The interledger approaches discussed in this survey include 1) atomic cross-chain transactions, 2) transactions across a network of payment channels, 3) the W3C Interledger Protocol (ILP), 4) bridging, 5) sidechains, and 6) ledger-of-ledgers. The approaches are compared according to whether they support the transfer or the exchange of value, their interconnection trust mechanism, complexity, scalability, and transaction cost.

INDEX TERMS Atomic swaps, blockchain, cross-chain transactions, distributed ledger technologies (DLTs), hash-locks, interledger protocol (ILP), sidechains, time-locks.

I. INTRODUCTION

Blockchain technologies [1], [2] have recently attracted massive research and business attention. What makes blockchains a disruptive technology is that they offer, for the first time ever, a tamper-proof append-only database where trust emerges through the collaboration of a set of mutually non-trusting computers, rather than through an institution or organisation that imposes trust from the external world onto the system. In fact, the respective trust guarantees are so strong that blockchains are suitable for storing and maintaining value ownership and transfer records, akin to banks, forming the novel application domain of *cryptocurrencies*.

In addition to Bitcoin, the pioneering and clearly dominant cryptocurrency to date, an overwhelmingly large number of alternative cryptocurrencies (often referred to as *altcoins*) have emerged [3]. Although at a high level they all serve a similar goal, significant tradeoffs distinguish them from each other [4], [5].

For instance, key tradeoffs exist with respect to the time it takes for a transaction to be committed and the security provided by a blockchain [6]. Other tradeoffs concern the level of privacy, with some blockchains being completely open to the public (referred to as *permissionless*) and others being deployed on authenticated servers exclusively (known as *permissioned*). The cost of operation constitutes another vastly differentiating factor between blockchains, with Bitcoin being reported to consume as high power as the entire country of Denmark, while other blockchains (mainly permissioned ones) can have negligible operational costs. Finally, blockchains differ significantly with respect to the capabilities offered by their *smart contracts*, as well as with respect to the way the execution of smart contracts is charged.

The aforementioned tradeoffs make it clear that the "one chain rules them all" paradigm is far from true. Instead, a wide spectrum of diverse blockchains are expected to continue operating in parallel, while we are very likely to witness the introduction of novel blockchains with currently unimaginable features. Finding ways to securely and efficiently interconnect such diverse blockchains becomes of paramount importance for guaranteeing a universal, unified, and non-segregated realm for distributed ledgers. This paper presents a survey and comparison of proposals for what is known as *interledger frameworks*.

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

The term *interledger* denotes a number of different approaches that attempt to establish interoperability among different distributed ledgers or blockchains.[1] The interledger approaches that have been proposed vary widely in their purpose and structure, ranging from *atomic cross-chain transactions* (or *atomic swaps*), which are based on *hash-lock* and *time-lock mechanisms* that either perform all or none of a cryptographically linked set of transactions, through the W3C *Interledger Protocol* (ILP), which is a TCP/IP-inspired architecture and protocol specification for routing digital assets through a network of payment channels, to *Polkadot* and *Cosmos* that effectively build *another ledger to interconnect different ledgers*.

A major shared motivation of all interledger proposals appears to be a push to move away from the ''one chain rules them all'' model, in order to increase the overall flexibility and innovation. In addition to this, there appear to be several other motivations for the various interledger approaches. We summarize these below and discuss them in more detail later in the paper.

- **Transferring and/or trading (or exchanging) value** between chains. With *transfer*, value is portable, i.e., it moves from one ledger to another. This is achieved by having the ''original'' value (tokens) in the first ledger frozen or locked (or destroyed) and the ''new'' value (tokens) in the other ledger unfrozen or unlocked (or created). With *trade* (or *exchange*), value (tokens) on different ledgers are exchanged simultaneously, i.e., the transactions that move value (tokens) from one account to another on the same ledger occur in an atomic manner. Unlike the transfer of value, the exchange of value is dependent on the exchange rate of the tokens being traded.
- **Transferring information** or generic messages between chains, in a way that the information or messages on different chains are cryptographically linked. This is particularly useful in Internet of Things (IoT) applications to immutably record information on multiple ledgers in a manner that satisfies some dependency conditions.
- Allowing a **different tradeoff between trust and cost** of the blockchain ecosystem. Typically, a higher level of trust necessitates a larger network with more peers or a more demanding consensus mechanism, thereby requiring a higher overall computation cost and leading to longer transaction confirmation times.
- Different levels of **privacy**. Permissionless blockchains, commonly referred to as public blockchains, such as Bitcoin and Ethereum,[2] allow anyone to participate in their operation and view the records stored on the ledger. At the other extreme, permissioned blockchains involve the collaboration of peers that belong to a specific (permissioned) set and can arrange their records to be opaque to others (private), or public (but only allow the permissioned set to contribute to the chain). Thus, permissioned blockchains can support different levels of write and read access, which allows them to support different levels of privacy.
- **Increasing the overall scalability and functionality**, in addition to **facilitating innovation** of the blockchain ecosystem. Multiple interconnected ledgers can exploit transaction locality to achieve scalability, while different ledgers can be designed to offer different functionality. Moreover, ledgers (e.g., sidechains) can be used to safely experiment with novel mechanisms, without influencing the properties and functionality of other ledgers (the ''main chains'').

A major difference between the various approaches is how the overall immutable state is assumed to be formed and maintained. In some approaches, such as atomic cross-chain trading, which utilizes hash-lock and time-lock mechanisms, the immutable state is stored only in the ledgers being interconnected. This is also the case with ILP. In other approaches, such as Cosmos and Polkadot, there is an attempt to establish a ''super-chain'' that verifies the consensus on various ''subchains''. The subchains are called *zones* in Cosmos and *parachains* in Polkadot. In still other approaches, the cross-chain immutability assumptions are relaxed, essentially creating two-phase transactions that either get confirmed or expire.

We have divided the various interledger approaches into the following categories:

1) Atomic cross-chain transactions
2) Transactions across a network: Lightning and Raiden
3) Layered value transfer protocols (W3C ILP)
4) Bridging approaches
5) Sidechains
6) Ledger-of-ledgers approaches

Different categories can use the same basic mechanism; for example, atomic swaps based on Hashed Time-Lock Contracts (HTLCs) are used in atomic cross-chain transactions for direct trading between two peers, in transactions-across-a-network (also referred to as *payment networks*), ILP, and some bridging solutions. Hence, the difference between the categories with respect to their underlying mechanisms is not always absolute. However, at a higher-level the various categories differ in their initial application assumptions. Atomic cross-chain transactions target peer-to-peer trading between two parties that seek to exchange value. Transactions-across-a-network solutions and ILP generalize peer-to-peer transactions to payment networks, where payments are routed along paths that are comprised of off-chain payment channels. Bridging approaches target cross-chain transactions between existing ledgers. Sidechain approaches assume the existence of a main chain and support the transfer of value between the main chain and sidechains, which are regarded as subordinate to the main chain. Ledger-of-ledgers approaches introduce a new super-ledger with the

[1] We will use the terms ''ledger'' and ''blockchain'' or simply ''chain'' interchangeably, noting however that distributed ledgers are a general category of ledgers that includes both blockchains and structures that are not based on cryptographically linked blocks that form a chain.

[2] We refer here to the Bitcoin and Ethereum mainnet, i.e., the open public instances of these blockchains and not private blockchains that could be deployed based on these technologies.

goal of having multiple sidechain-like ledgers, which can also support the interconnection to existing ledgers, such as Ethereum and Bitcoin.

The approaches we discuss can be applied for interconnecting both permissionless (or public) and permissioned (or private) ledgers. A difference is that for permissioned ledgers the nodes or modules that perform the interconnection would need to obtain the necessary credentials to submit transactions. Moreover, permissioned ledgers might have constraints or restrictions that need to be adhered to. Also, permissioned ledgers typically do not have a native token. However, permissioned ledgers can digitally record the ownership of assets, which can be used in the exchange or transfer of value across chains. Finally, in the sidechain and ledger-of-ledgers approaches, the central or main chain is typically a public ledger, whereas the sidechains can be permissioned ledgers having lower transaction cost and delay, but higher privacy.

The various interledger approaches are compared in terms of the following features: i) whether they support the transfer of value or the exchange of value, ii) the interconnection trust mechanism, iii) complexity, iv) scalability, and v) transaction cost. Approaches that perform the exchange of value across two or more chains rely on the consensus mechanisms of the chains that are involved, which provides decentralized trust, thus avoiding the need for a single trusted entity. The interconnected trust mechanism defines where the immutable state of the transactions across chains is recorded; this is related to the mechanism which ensures the trusted execution of these transactions, without relying on a single trusted entity. The complexity of the interledger approach is determined by the amount of data (transactions) from each of the interconnected chains that the approach needs to process in order to ensure trusted commitment of transactions across chains. Scalability refers to the total number of transactions that a solution can support per unit of time, and how the incremental cost for supporting additional transactions depends on the total number of transactions per unit of time. Finally, the transaction cost refers to the aggregate cost of all transactions, which depends on the percentage of transactions that are inside the main chain or inside the sidechains and the transactions that are across the two.

The remainder of this paper is structured as follows. We first discuss related surveys in Section II, identifying how they differ from the current survey. Next, we cover each of the interledger approaches separately in corresponding sections. Since atomic cross-chain transactions are fundamental building blocks used by some of the other approaches, we start with their discussion in Section III. We discuss transactions-across-a-network approaches in Section IV and W3C's ILP in Section V, which can utilize atomic cross-chain transactions and do not involve the introduction of an interconnection ledger or new functionality inside an existing ledger. Next, we discuss bridging approaches in Section VI. Finally, we discuss sidechains in Section VI followed by ledger-of-ledgers approaches in Section VIII.

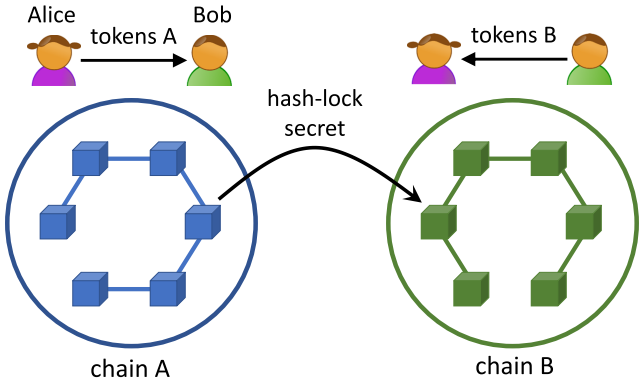


FIGURE 1. Atomic cross-chain trading. Having the same hash-lock value in the transactions on the two chains can ensure that either both of the two transactions (Alice and Bob exchanging tokens on the two blockchains) occur, once the hash-lock secret is revealed, or neither occurs.

II. RELATED WORK AND SURVEYS

The focus of this survey is interledger approaches, which differentiates it from other blockchain-related surveys that we mention next. Christidis and Devetsikiotis [7] discuss the underlying mechanisms of blockchains and smart contracts, identifying the features and issues that arise when blockchains and smart contracts are applied to the Internet of Things (IoT). Yeow *et al.* [8] present a survey of decentralized consensus systems for edge-centric IoT, focusing on their data structure, scalability, and transaction model. Ali *et al.* [9] survey recent state-of-the-art efforts investigating the application of blockchain technology to provide a decentralized, trustless, and secure environment for the IoT. Xu *et al.* [10] present a taxonomy of blockchain systems according to the level of decentralization, verification model, storage and computation, protocol configuration, and support for sidechains or multiple blockchain deployments. Tasca and Tessone [4] present a taxonomy tree of blockchain technologies based on building blocks related to the consensus model, transaction capabilities, native currency, and extensibility, which includes interoperability with external systems and with other blockchains. NIST's report by Yaga *et al.* [1] presents a high-level technical overview of blockchain technology, identifying their basic components and discussing consensus models. The work by Tschorsch and Scheuermann [11] focuses specifically on the Bitcoin protocol and its building blocks. Finally, Li *et al.* [12] present a survey of security threats and corresponding real attacks on blockchain systems.

At the time of this writing there are very few academic peer-reviewed works in the inter-blockchain and interledger areas. In a recent paper, Herlihy [13] presents the first comprehensive work that analyses the basic atomic cross-chain swap mechanism, which is discussed in Section III. Back *et al.* [14], in their working paper frozen in October 2014, explain the background and the basic ideas of sidechains. Sidechains are discussed in Section VII. Chen *et al.* [15] give an incomplete idea for a *Byzantine Fault*

Tolerance (BFT)-based ledger-of-ledgers approach, without discussing the other approaches in the field, e.g., ILP and Polkadot. Croman *et al.* [16] discuss blockchain scalability in general terms and mention sidechains and off-chain transactions as two possible approaches to address scalability. Dilley *et al.*, in another unpublished paper [17], propose strong federations, which consider a byzantine layer on top of multiple blockchains; these are discussed in Section VII-A. English, Orlandi, and Aueris, in yet another preprint [18], describe the Uberledger framework, which is a type of ledger-of-ledgers approach and is discussed in Section VIII. Sun *et al.* [19] describe Multi-Blockchain Digital Currency (MBDC), a permissioned blockchain technology making use of a multi-blockchain architecture, targeting central bank transaction systems. Tate, Johnstone, and Fielt [20] briefly mention inter-blockchain communication using as an example Cosmos, which is discussed in Section 5, indicating that it may allow users to transfer their reputation between blockchains. In another technical report, Buterin [21] presents an overview of three strategies for blockchain interoperability: centralized or multi-sig notary schemes, sidechains/relays, and hash-locks and time-locks.

III. ATOMIC CROSS-CHAIN TRANSACTIONS

Atomic cross-chain transactions focus on the basic problem of trading assets on two unrelated blockchains. Specifically, two parties, Alice and Bob, wish to trade digital assets on two blockchains, A and B, on which they both have accounts: Alice wants to give user Bob some amount of assets on chain A in exchange for some amount of assets on chain B, that are owned by Bob. Note that such an exchange involves *trading* of digital assets rather than actually *moving* digital assets from one blockchain to the other. Such a trade entails risks, since the user who first gives the other user the agreed amount of assets is faced with the risk of the other user not obeying the agreement and keeping both the assets he received from the first user, as well as the assets he promised to give. Indeed, the immutable nature of blockchains aggravates this issue, since transactions recorded on a blockchain cannot be revoked.

In the financial sector, the aforementioned risk is handled by the so-called *Delivery-versus-Payment* (or *Payment-versus-Payment* if the trade involves currency assets) procedure, which requires the presence of a trusted third party that ensures that either both transfers occur or neither does. Hence, one approach for performing transactions across chains involves a trusted third party that ensures the trade is atomic. Instead of a single party, the assurance of atomicity can be provided by a multi-sig[3] notary scheme or a group of parties (federation).

Another approach for trading digital assets across chains is atomic swaps [21], also known as atomic cross-chain trading [22], [23]; see Figure 1. This approach involves publishing transactions on the two blockchains utilizing hash-locks and time-locks in a way that atomicity is ensured: either both transactions take place or neither takes place. The above atomicity is achieved without requiring a trusted third party. Moreover, the users involved in the trade do not need to trust each other.

Atomic swaps are based on *Hashed Time-Lock Contracts* (HTLC) [24], which utilize the following basic mechanisms:

- Multi-signatures: transactions can be signed by two (or more) parties, thus having the parties that signed verify and be accountable for the multi-signature transaction (also referred to as multi-sig in Bitcoin parlance).
- Hash-locks [25]: a cryptographic lock that can be unlocked by revealing a secret s whose hash $H(s)$ is equal to the value h configured in the lock.
- Time-locks [26]: a time-based condition that prevents a transaction's or smart contract's assets from being redeemed (or refunded) until a specific time interval has elapsed. The interval can be relative to the time the transaction is published on the blockchain or can be absolute time.
- Basic scripting: this is required to indicate that a transaction can be unlocked (or committed) only if multiple conditions are satisfied, e.g., both the secret to unlock a hash-lock is revealed or the time specified by the time-lock has elapsed and a particular signature is provided.

Hash-locks can be used to link transactions on two blockchains. Both locks are constructed using the same hash function and are configured with the same hash, h, hence they can be unlocked using the same secret: opening one hash-lock reveals the secret which can be used to open the hash-lock on the other chain. Specifically [22], [23], Alice selects a random number s and submits a transaction on chain A according to which some amount of her tokens on chain A are transferred to Bob's account on that chain; this transaction has a hash-lock with value $h = H(s)$, thus is executed only if the secret s, initially known only by Alice, is submitted to chain A. Similarly, Bob submits a transaction on chain B according to which some amount of his tokens on chain B are transferred to Alice's account on that chain; this transaction has the same hash-lock as the first transaction on chain A. Alice can reveal the secret s on chain B to obtain the tokens on that chain. Once the secret s is revealed, Bob can submit it to chain A to obtain the tokens on that chain, according to the first transaction. If Alice never reveals the secret s, then the tokens on both chains would be locked indefinitely. To avoid this situation, two additional transactions are submitted to the two chains: the additional transaction on chain A allows Alice to redeem the tokens she locked in the first transaction only after time T_1 has elapsed, which is implemented using time-locks. The second transaction on chain B allows Bob to redeem the tokens he locked in the first transaction on chain B only after time T_2 has elapsed. To avoid the case where Alice obtains the tokens on chain B and also redeems the tokens on

[3]Multi-signature (multi-sig) transactions are explained and discussed later in this paper.

chain A, T_1 must be larger than T_2. Hence, Alice must submit the secret s on chain B to obtain the tokens on that chain until time T_2. Once Alice reveals the secret s, Bob must submit the secret to chain A until time $T_1 > T_2$ in order to obtain the tokens on that chain.

Hence, trust is ensured through atomicity of the two linked transactions that transfer tokens from Alice to Bob (chain A) and from Bob to Alice (chain B) and by immutably recording the state involving the value exchange on the two ledgers. Other than the above basic mechanisms, no modifications to the two chains are necessary. Of course, the two parties performing the exchange must have wallets on both chains.

Because atomic swaps are based on hashes and comparisons, they have lower complexity compared to the other approaches that we discuss in the following sections. Moreover, they require basic scripting capabilities rather than the more advanced smart contract capabilities available in blockchains such as Ethereum, hence they can be supported by blockchains such as Bitcoin, which do not support smart contracts. Because atomic swaps involve simple operations, the risk of a mistake is lower. Also, two chains supporting cross-chain transactions with atomic swaps can have higher scalability compared to a single chain, in the presence of locality that reduces the number of transactions across the two chains. This occurs because the whole ledger is replicated and processed by all blockchain nodes. Hence, in the presence of locality, the blockchain nodes belonging to one of the two chains would replicate and process a smaller ledger that contains only the transactions on that chain. On the other hand, in the case of a single chain with the same total number of transactions as in the two chains, blockchain nodes would need to replicate and process a ledger containing all the transactions.

Although atomic swaps have low complexity, their cost involves the cost of the transactions on the two ledgers. This cost can be reduced with payment channels, discussed in Section IV, which focuses on transactions across a network and in Section V, which focuses on WRC's ILP. One requirement for achieving the atomicity of atomic swaps is that, once the secret for the hash-lock is revealed on one chain, some entity obtains the secret and submits it to the second chain. This action can be performed by either of the two parties that are exchanging value. Indeed, the two parties have the incentive to submit the secret, since if the secret is not submitted on the second chain then one of the parties can lose the amount they are exchanging. Alternatively, the secret can be submitted by some interledger gateway. Atomic cross-chain transactions are employed by some bridging proposals, which utilize decentralized trust algorithms to ensure that the transactions are submitted on the involved chains and to provide additional functionality, such as discovery and registration; such solutions are discussed in Section VI.

When atomic swaps are performed between different chain technologies, proper timing of the time-locks on the two chains may be an issue. Specifically, when the two chains have block mining or confirmation times with high variability, this needs to be taken into account when defining the duration of the time-locks on the two chains. In any case, the completion time of an atomic swap is bound by the chain with the slowest block time. Because atomic swaps involve the exchange of digital assets, they are dependent on the exchange rate of assets. This exchange rate can fluctuate during the execution of an atomic swap, whose duration depends on the block time of the chains involved. Although this can be seen as a disadvantage [21], it provides financial call options [27]. Call options arise when one of the two trading parties must act first and the other must follow. Hence, the second party has the option to complete or abort the trade; the decision which option to select can depend on the price changes of the assets exchanged in the intervening period, which depends on the transaction time-locks.

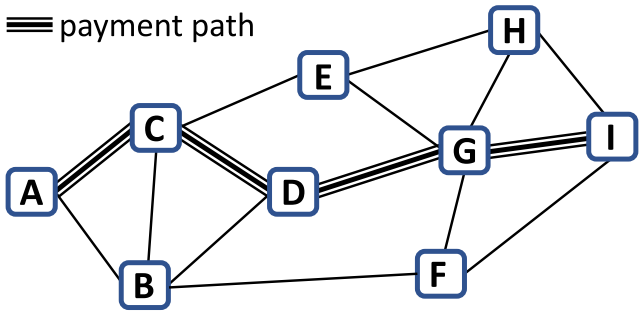


FIGURE 2. Nodes interconnected with a link are entities that have established a payment channel. In the Lightning network a payment path consists of a series of payment channels, which can involve different currencies of different blockchains.

Blockchains provide an immutable recording of transactions based on distributed trust. However, cross-chain trading is still mostly performed by (single) third parties, i.e., in a centralized fashion. Atomic cross-chain protocols enable trusted peer-to-peer trading eliminating the need for third parties. Hence, an important application of atomic swaps is the support for decentralization and decentralized exchanges. However, a requirement is that the parties trading assets must be online until the trade is completed.

Moreover, although atomic swaps are typically used for cross-chain trading, they can also be applied on a single chain to obfuscate the transaction graph, hence to increase privacy.

Atomic swaps have been performed between different cryptocurrencies, initially between Decred and Litecoin in September 2017, between Ethereum and Bitcoin in October 2017, and between many other cryptocurrency pairs ever since. HTLCs are also used in constructs such as the Lightning Network and for improving the scalability of Bitcoin by enabling off-chain transactions between untrusted parties [28], [29].

Atomic swaps are well-known in the blockchain community, but the only systematic analysis of their properties is from Herlihy [13], who provides an analysis of atomic swaps when multiple parties exchange assets. Specifically, if cross-chain atomic swaps are modelled as a directed graph, an atomic swap protocol based on hash and time-locks is

possible in a system with rational parties only if the directed graph is strongly connected; if the graph is not strongly connected, then rational parties will not agree to a swap because of the existence of freeriders. Moreover, Herlihy [13] shows that an atomic swap protocol has time complexity proportional to the graph's diameter and communication complexity proportional to the number of value exchanges.

IV. TRANSACTIONS ACROSS A NETWORK: LIGHTNING AND RAIDEN

The solutions in this section provide a decentralized system for routing micropayments through an interconnection of micropayment channels, which realize the off-chain exchange of value [28]. Micropayment channels are two-party accounts which contain an initial deposit made by the two parties. To spend funds of the channel, both parties need to agree on the new balance. The agreement between the two parties can be performed off-chain, thus it does not incur the cost for committing transactions on the blockchain. Micropayment channels can be seen as a second layer on top of the blockchain, which is considered the first layer.

The Lightning Network [29] is a decentralized system for a sender to send a Bitcoin payment to a receiver, where the Bitcoin transactions are sent over a network of micropayment channels whose transfer of value occurs off-chain. In each payment channel, Bitcoin transactions are signed by the corresponding two parties. The transfer of value takes place between untrusted parties along the payment path to the final payment receiver. Micropayment channels are bi-directional and utilize HTLCs. Specifically, any two parties, say Alice and Bob, may agree to both store some bitcoin assets to a micropayment channel by publishing a funding transaction in the Bitcoin blockchain. While moving funds in Lightning, both Alice and Bob always also have a latest commitment transaction, signed by both parties, which either may opt to unilaterally publish in the Bitcoin blockchain. In practice, the commitment transaction contains two distinct transactions: One where Alice releases Bob's share immediately but where Alice's share is time-locked and revocable by Bob, if Bob knows the revocation key, and vice versa. When performing a transaction in Lightning that involves generating a new pair of commitment transactions, the revocation keys of the previous commitment transaction are revealed; since Alice and Bob know the previous revocation keys, they both have an incentive to conform to the protocol and act in a trustworthy manner. Only the initial transaction, which contains the deposit from both parties to the channel, and the final transaction that closes the channel are published on the blockchain. All intermediate commitment transactions are directly transferred off-chain between the two parties in a peer-to-peer fashion.

One significant benefit of utilizing off-chain transactions is the reduction of the total transaction cost, which can be significant if the number of peer-to-peer transactions is high.

Lightning depends on the Segwit Bitcoin extension, which was a part of the Bitcoin Elements library. The extension was activated on Bitcoin in August 2017 and has gained reasonable usage since then. Hence, Lightning appears to be a fully functional part of the Bitcoin ecosystem, undergoing active development in the community. The Lightning Network core is a separate small group of people, coordinating the maintenance of the Lightning daemon[4] and protocol specification at Github.[5]

Similar to the Lightning Network, the Raiden Network[6] supports off-chain transactions and the transfer of value across a network of interconnected payment channels. Similar to the Lightning Network, Raiden Network's most basic building block is the payment channel, whose off-chain transactions incur zero fees. At the end of 2017, a simplified version of Raiden called μRaiden[7] was activated on the Ethereum mainnet. μRaiden supports off-chain token transfers to predetermined receivers, utilizing payment channels. μRaiden does not support multihop token exchanges.

Another effort that is based on off-chain payment channels and supports multiple assets is COMIT,[8] where liquidity providers operate between two or more blockchains acting as market makers in a decentralized exchange marketplace [30].

In addition to the reduced cost, off-chain transactions help improve scalability. Scalability can also be improved in a similar manner to the way the Internet achieves scalability, by distributing the transactions in the whole network across intermediate nodes that route payments from the initiating to the receiving party. Indeed, the payment channels that define the path from the initiating to the receiving party can be on different blockchains, hence can involve different currencies which also necessitates the exchange between currencies. This feature is further highlighted with the Interledger Protocol (ILP), which we discuss in the next section.

Although payment channels and networks reduce cost and improve scalability, they require a routing procedure to determine the payment channels that create a path from the initiating party to the receiving party; such a routing procedure needs to consider the funds available in the payment channels. Initial routing proposals, such as [31], considered static routing, where the route is defined by the initiator. One limitation of static routing is that it cannot adapt to varying network conditions. Work such as [32] investigates more dynamic schemes for routing in payment networks, utilizing path probing to identify paths containing channels with sufficient funds. Another important issue is privacy, since with simple source routing the full path is included in the payment request along the whole path, hence is visible by all intermediate nodes. The Lightning network is addressing this with an onion routing scheme [33], where any node along the payment path knows only its predecessor and successor. More recent work has proposed anonymous multi-hop locks

[4]https://github.com/lightningnetwork/lnd
[5]https://github.com/lightningnetwork/lightning-rfc
[6]https://raiden.network/faq.html
[7]https://microraiden.readthedocs.io/en/latest/
[8]https://www.comit.network/

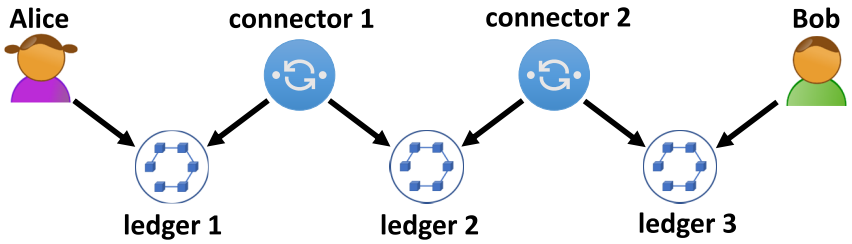


FIGURE 3. ILPv1 transaction between Alice (account in Ledger 1) and Bob (account in Ledger 3) via Ledger 2, with the help of two connectors.

for payment networks with improved security and privacy guarantees [34].

One limitation of payment channels is that they require that funds be locked in the shared channel account from the time the channel is opened until it is closed. The work in [35] proposes a system for rapidly changing the allocation of funds to channels, without requiring opening new channels. Moreover, the problem of balancing the funds available in network channels is inter-related with the procedure for routing payments across the network and the fees that intermediate nodes apply for processing payments [36].

V. W3C INTERLEDGER PROTOCOL (ILP)

The World Wide Web Consortium (W3C) proposed a generic protocol to enable the secure transfer of funds across any two ledgers. The protocol, known as the *Interledger Protocol* (*ILP*) [37], [38], has undergone a number of design changes, with its key implementations being version 1 (*ILPv1*) and version 4 (*ILPv4*).

The goal of ILP is to allow the *atomic* transfer of funds from a ledger A to a ledger B, in such a way that no involved party incurs any risks, and such that the sender can have an indisputable proof that the final receiver redeemed the respective funds (i.e., the *non-repudiation* property).

Most of the observations for ILP in terms of complexity, scalability, and transaction cost are the same as for the transaction-across-a-network approach, since both can use HTLCs and payments are routed across a network of payment channels. The main difference is that ILP focuses on defining an open protocol for the interconnection of ledgers.

A. ILPv1

ILPv1 is described in the respective white paper [37]. It leverages escrow transactions to transfer value among accounts on different ledgers.

Let us assume that a user referred to as the *sender* wants to transfer value to a user referred to as the *receiver*. Had they both accounts in the same ledger, that would be straightforward through a simple transaction. In our scenario, however, they maintain accounts in separate ledgers, A and B, respectively. The transfer can be facilitated by a third user, referred to as the *connector*, who maintains accounts in both ledgers A and B. The idea is that the sender will transfer value to the connector in ledger A, and the connector will transfer the respective amount to the recipient in ledger B. The amount reaching the recipient will depend on the amount paid by the sender, as well as the connector's exchange rate for conversions from A to B and its service fee.

A complication arises from the fact that this transfer involves two distinct transactions. If the sender makes a transfer to the connector first, he/she has to trust that the connector will also do his part, transferring the respective value to the recipient. Likewise, if the connector transfers the amount to the recipient first, he has to trust that the sender will pay him accordingly. ILP resolves this issue by making the entire transfer atomic, that is, either both transactions are executed or none. This is achieved with the help of *escrow transactions*, that is, transactions whose redemption requires the satisfaction of a condition.

More specifically, the end-to-end value transfer takes place as follows. First, the sender's payment to the connector is registered in ledger A. Subsequently, the connector's payment to the recipient is registered in ledger B.

Both transactions are escrowed pending the recipient's signature on some arbitrary transaction identifier set by the sender. They are, thus, escrowed on the same condition. At this point, the only one who can redeem funds is the recipient, as he/she can provide the required signature to the transaction in ledger B. The recipient's signature unlocks his/her funds, but at the same time also reveals the necessary signature to unlock other transaction escrowed on the same condition. Hence, the connector can also unlock the funds sent to it by the sender in ledger A.

In other words, transactions (in ledger) A and B are registered in this order, and are escrowed in such a way that transaction A can be redeemed *if and only if* transaction B has already been redeemed. This way, all involved parties are protected. For the recipient this is obvious: he never was at risk of losing anything. As far as the connector is concerned, he knows that as soon as his payment to the recipient has been redeemed, he will also be able to redeem the sender's payment to him. Finally, the sender can rest assured that unless the recipient has been paid, the connector will not be able to redeem his payment.

This scheme can be easily extended to any number of intermediate connectors, in case no single connector exists that directly links ledgers A and B. In the general case, transferring value from ledger L_1 to ledger L_k can be realized with the assistance of intermediate ledgers L_i, for $i \in \mathbb{N} \cap [2, k-1]$, and respective connectors $C_{1\to 2}$, $C_{2\to 3}$, $\ldots$, $C_{k-1\to k}$.

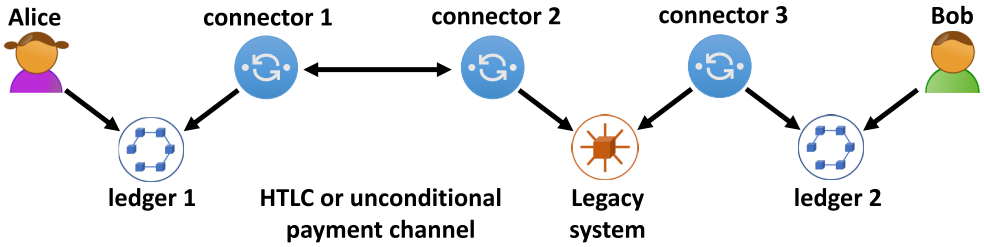


FIGURE 4. ILPv4 transaction between Alice and Bob with the help of three connectors. The connectors can use on-chain transactions, HTLC and unconditional payment channels, and legacy payment systems.

Figure 3 shows an example with three ledgers and two connectors.

To prevent having funds being kept in escrow forever, in case the recipient opts not to redeem its assets, timeouts are set in the escrow transactions. The assets sent by the connector can be collected by the connector itself after a timeout T_c has elapsed, and similarly the sender can collect its own assets back after a timeout of T_s. Clearly, T_s should be set sufficiently later than T_c, to prevent having the sender collect its initial payment back, and *then* having the recipient redeem the collector's payment.

B. ILPv4

The initial interledger protocol, ILPv1, suffered a number of shortcomings, mostly stemming from the long timeouts inherent in escrow transactions, typically spanning one or more *days*. In an attack coined as the *free option problem*, a colluding sender and receiver could set up a transfer, effectively committing the participating connectors to a certain exchange rate, and then wait until just before the timeout expires to decide whether to proceed with redeeming the transaction if the actual exchange rate changed in their favor, or to let it time out otherwise.

On another attack vector, a malicious sender could set up a bogus transfer to itself knowing it will fail, with the intention to tie up intermediate connectors' funds for the timeout duration, essentially performing a *denial of service* attack with respect to connectors' liquidity.

To alleviate risks associated with slow transfers and with high-value commitments, ILPv4 is designed around fast transfers of low-value packets [39]. Ledger-based escrow payments are inherently slow and expensive, thus they are dropped from being a requirement for value transfers. Instead, ILPv4 allows connectors to set up bilateral trust relations of arbitrary type. Although this appears to call for high risks compared to escrow payments, all risks are confined exclusively between directly interacting connectors, let alone they typically concern small values. Importantly, senders and receivers enjoy a completely risk-free operation.

More specifically, directly interacting entities (i.e., sender and first connector, two consecutive connectors, or last connector and receiver) have to maintain an account with each other on a settlement system of their choice. Such a system could be a ledger, as is the case in ILPv1, but it is not enforced. Quite interestingly, it is discouraged, as a ledger would have a detrimental effect on transfer speed. Instead, unidirectional or bidirectional payment channels, off-chain transactions, and side chains could be used as a settlement system between two interacting entities. Indeed, rather than HTLC payment channels, *unconditional payment channels* with short timeouts can be used. Also, traditional channels such as credit cards, bank accounts, Paypal, or even cash could do as well, although this would decrease the benefit of using ledgers in the first place.

A bilateral trust relation depends on the two involved entities only. It can be a post-funded agreement, in which the payer should settle his debt to the payee every time some credit limit is reached. It can be a pre-funded agreement, in which the payer deposits some funds in advance. Or it can be based on a ledger with escrow payments (slow and expensive).

In ILPv4 transfer of value is carried out in two phases, *prepare* and *fulfill*. Initially, the sender generates a *prepare* packet containing the value being sent and the hash $H(s)$ of a secret s known by the sender and the receiver only. The prepare packet is routed to the receiver through one or more connectors. Forwarding this packet to the next connector towards the target represents your commitment to pay him *if and only if* he presents proof that he already paid *his* next connector (or the receiver). Once the receiver receives the prepare packet, he generates a *fulfill* packet revealing the secret s, and sends it back to the sender in the reverse path, starting with the last connector. When the last connector receives it, it validates that $H(s)$ indeed corresponds to s, makes the payment to the receiver, and forwards it to the previous connector. Payments are made one-by-one, till the fulfill packet reaches the sender, who then pays the first connector.

Note that it is connectors rather than ledger scripts that check the fulfill condition. This allows the prepare and fulfill packets to propagate fast (in the order of seconds) between the sender and the receiver, rather than enduring the day-long delays of escrow payments.

VI. BRIDGING APPROACHES

Bridging refers to approaches that aim to provide one or two-way transfer or exchange of both value and information between blockchains that are considered more-or-less "equal". In this respect they differ both from sidechains and

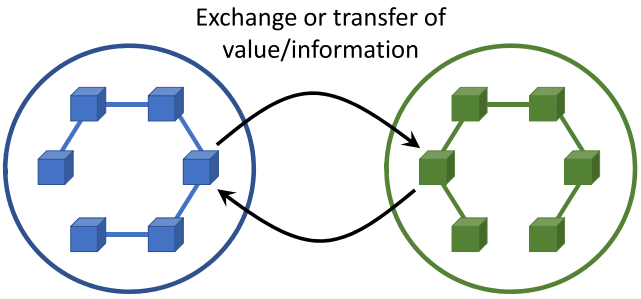


FIGURE 5. Bridging approaches typically involve modules running on the nodes of the two (or in some cases only one) interconnected chains that are used for exchanging or transferring value or information between the two chains.

ledger-of-ledgers approaches that involve a main chain interconnected with one or more sidechains, which are regarded as subordinate to the main chain. Bridging approaches typically involve modules or smart contracts running on nodes participating in both (or in some cases in only one) of the interconnected chains. The contracts are used for monitoring the transactions and for exchanging information between the two chains; see Figure 5.

Some bridging solutions utilize atomic swaps, thus support the exchange of value, but provide additional functionality, such as service discovery and registration. In this respect they differ from both atomic cross-chain transactions with peer-to-peer interaction of two parties and ledger-of-ledgers approaches. Other bridging solutions are based on Ethereum smart contracts and support the transfer of value. Bridging approaches have a high computation cost when they require nodes to view and process the entire blockchain of the interconnected ledgers. The bridging proposals that we discuss below are Blocknet, ARK, BTC Relay, the POA network, Wanchain, and Aion.

Blocknet[9] was launched in 2014 with an aim to be "The Internet of Blockchains" [40]. It is founded on a protocol called XBridge, a peer-to-peer protocol that aims to enable communication between nodes on different blockchains.

Blocknet's main product is a decentralized exchange that allows any Bitcoin-like cryptocurrencies to be exchanged without a centralized party, as long as the currencies involved support BIP 65 (CheckLockTimeVerify), which has been in Bitcoin since late 2015. Blocknet's goals are similar to ILP's, although Blocknet is designed only for cryptocurrencies, while ILP aims to address the exchange of other types of value in addition to cryptocurrencies.

Blocknet is comprised of two core components: XBridge and XRouter. XBridge provides a DHT-based peer-to-peer network, which acts as an inter-chain network overlay. XRouter provides service lookup and registry services that are necessary to route inter-chain messages to the correct blockchain. Cross-chain transactions are based on HTLCs, combined with a protocol for verifying that the service node implementing the cross-chain transaction is the claimed provider. Service nodes along with staking nodes implement a Proof-of-Stake consensus algorithm to ensure decentralized trust. The interconnected blockchains can be both permissionless and permissioned ledgers.

ARK[10] is another system that markets itself as a bridge [41], being somewhat similar to Blocknet. That is, ARK's so called Smart Bridges are similar to Blocknet's XBridge since they connect distinct blockchains and facilitate communication between them. However, ARK acts as the intermediary between different chains using a Delegated Proof-of-Stake (DPoS) consensus algorithm. ARK allows existing and new blockchains to communicate with each other in order to support more than token swaps, such as to execute service contracts, which can include the transfer of data, the creation of smart contracts, and the execution of code on blockchain platforms.[11]

To achieve compatibility with another blockchain, a small portion of code needs to be inserted into the core of the blockchain. This allows the blockchain to interact with ARK.

Recall that atomic cross-chain transactions discussed in Section III allow peer-to-peer trading between two parties. However, this requires that both parties are online until the exchange is completed. One application of bridging approaches such as Blocknet and ARK is to provide a decentralized exchange that does not require the trading parties to be online. The decentralized exchange is implemented by nodes through a distributed consensus algorithm, thus providing a decentralized trust system for reliably executing trading transactions.

BTC Relay,[12] which was initiated by the Ethereum Foundation, is a smart contract on Ethereum that can read the Bitcoin chain and verify Bitcoin transactions. This allows using Bitcoin payments for executing Ethereum smart contracts. BTC Relay was released in early 2016, being the first Ethereum-Bitcoin cross-chain production implementation [21]. BTC Relay uses Bitcoin block headers to build a "mini-version" of the Bitcoin blockchain. When an application processes a Bitcoin payment, it uses a header to verify that the payment is legitimate. Relayers are those who submit block headers to BTC Relay. When any transaction is verified in the block, or the header is retrieved, relayers are rewarded with a fee. Note that the interoperability supported by BTC Relay is one-way: Bitcoin cannot read the Ethereum chain, because its scripting language is not sophisticated enough.

The POA Network[13] utilizes Proof-of-Authority (PoA) as its consensus mechanism and is another attempt for developing a cross-chain bridge solution for connecting Ethereum-compatible blockchains. The POA Network is based on the Parity bridge open-source project.[14] POA provides developers with the flexibility to code in Ethereum

[9] http://blocknet.co

[10] https://ark.io
[11] https://arkaces.com/services/
[12] http://btc-relay.readthedocs.io
[13] https://poa.network
[14] https://github.com/paritytech/parity-bridge

standards while being able to utilize POA Network's solutions, such as the POA Bridge for interoperability between blockchain networks. The POA Bridge is an interoperability protocol where users can transfer value (ERC-20[15] compatible tokens and POA network coins) between permissioned chains that are based on PoA consensus and the Ethereum network. The POA bridge operates by locking POA coins on the POA network side and minting ERC-20 tokens on the Ethereum network.

The proposal by Wanchain [42] is an Ethereum-based generic ledger that supports cross-chain transactions using smart contracts, aiming at building a "distributed bank", where clients can transact using cryptocurrencies of their choice. Specifically, to perform a cross-chain transaction, tokens from the original chain (Ethereum, ERC-20, or Bitcoin) need to be transferred to an Ethereum account, which essentially locks the tokens being transferred. Once the original tokens are locked, a new smart contract is created which will handle the respective shadow representations of the original tokens within the Wanchain network. Cross-chain transactions are verified by verification nodes (called Vouchers) that implement a Proof-of-Stake consensus algorithm and receive transaction fees when they provide correct verification proofs; these fees are in Wanchain's native coin. In addition to smart contract-based cross-chain transactions, Wanchain supports the exchange of tokens between different blockchain systems, e.g., Bitcoin and Ethereum, using atomic swaps. Finally, transaction anonymity is guaranteed using the Ring Signature scheme [43].

Aion is a proposal that has common features to the proposals discussed above. Namely, inter-chain transactions are performed by bridges, which implement a lightweight BFT-based consensus algorithm and receive inter-chain transaction fees. The interconnected chains can be public or permissioned chains with their own governance, consensus, and participation rules [44]. A recent report describes a notary-based scheme with a well-defined trust model for message transfer between two smart contract-enabled blockchains [45].

The bridging approaches discussed in this section consider a consensus mechanism, such as Proof-of-Stake, Delegated Proof-of-Stake, or Proof-of-Authority among the nodes that perform the bridging functionality and can include paying fees to these bridging nodes for the interconnection services that they provide. These features are common to the ledger-of-ledgers approaches, which are covered in Section VIII. A distinction is that the target of bridging is to enable cross-chain transactions between existing ledgers, while the goal of ledger-of-ledgers approaches is to introduce a new super-ledger, having multiple sidechain-like ledgers.

[15]ERC-20 stands for Ethereum Request for Comments (ERC) 20 and is a standard for smart contracts on Ethereum for implementing tokens.

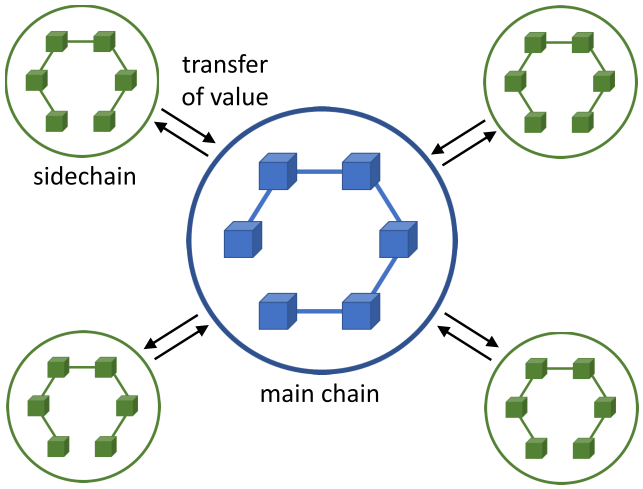


FIGURE 6. With the sidechains approach there is transfer of value to/from the main chain (or parent chain) and sidechains.

VII. SIDECHAINS

The basic idea of a sidechain is to move some assets from one blockchain, often called the *main* or *parent chain*, to one or more other chains, referred to as *sidechains*, in order to conduct some transactions there. Later on, the assets can be moved back to the original chain. A common motivation for using sidechains is transaction confirmation time. The transaction delays on the sidechain are typically much smaller than on the main chain. The main chain can be Bitcoin or Ethereum, with their 10 minutes or 20 seconds basic confirmation times, respectively (and longer times if higher security is required). A second reason to use sidechains is that a sidechain may support some functionality that the main chain may not have, e.g., the programming capability of smart contracts, or even experimental features. Finally, the transaction cost on the sidechain could be (significantly) lower than on the main chain.

The reduced transaction time and transaction cost can boost scalability. To achieve the aforementioned advantages, sidechains can be permissioned ledgers with their own governance and participation rules. On the other hand, the complexity of sidechain solutions can be high due to the high computation cost if they require code to view and process the entire blockchain of the interconnected ledgers; Simplified Payment Verification (SPV) proofs can reduce this cost.

Market forces can influence the joint operation of the main chain and its sidechains. While developing new blockchains and *alternative coins* is technically easy, creating a market for them is hard; a market is required for creating an incentive for mining. However, if the assets in a new blockchain can be securely bound to existing assets in a major blockchain, such as Bitcoin, many of these problems may be alleviated or circumvented. For example, it may be possible to issue transaction fees in a sidechain in such a manner that the sidechain miners can exchange them into main chain assets without any interaction from the other parties in the sidechain or the main chain.

The most typical sidechain logic involves the following steps:

1) Freezing some assets in the main chain in such a way that they can be unfrozen later.
2) Creating (or unfreezing) corresponding assets in the sidechain(s).
3) Performing transactions in the sidechain(s), perhaps moving assets further between two or more sidechains.
4) Deleting (or freezing) some or all of the created (or unfrozen) assets in the sidechain(s). This forms an agreement on how the assets are going to be further distributed in the main chain.
5) Unfreezing the assets in the main chain, moving them forward to one or more parties based on an agreement in the main chain that reflects the agreement in the sidechain.

While most of the above can be implemented technically in a relatively straightforward manner, the tricky part is freezing the assets in the main chain in such a way that they can be later unfrozen securely and be distributed based on the agreement(s) in the sidechain(s). The existing sidechain approaches differ from each other in who is trusted to unfreeze the assets in the main chain, the level of trust between the main chain and the sidechains, and the resolution procedure in case of disputes. For example, if the validators (miners) of the main chain are completely unaware of the sidechain, the freezing of assets in the main chain should be done in such a manner that the activity in the sidechain creates evidence that, when presented on the main chain, is considered as valid verification of possession that allows moving some or all of the frozen assets on the main chain. Furthermore, when more transactions are created in the sidechain, some of the older evidence generated in the sidechain should become invalid, as the assets have moved again.

Back *et al.* [14] define the following requirements for sidechains:

1) Assets should be able to be moved back to the main chain by whoever their current holder in the sidechain is, and nobody else (including previous holders).
2) There should be no ability for a dishonest party to prevent the transfer of assets from occurring.
3) Transfers should be atomic, i.e., they should happen entirely or not at all.
4) Sidechains should be firewalled: a bug in a sidechain enabling creation (or theft) of assets in that chain should not enable the creation or theft of assets on any other chain.
5) Blockchain reorganisations should be handled cleanly, even during transfers.
6) Users should not be required to track sidechains that they are not actively using.

While Back *et al.* [14] describe only a few - *at that time future* - possibilities for creating sidechains, the paper has inspired a number of commercial attempts to create and utilize them, including Rootstock, Blockstream sidechains, and the Lightning Network. We consider first the original federated pegs and Blockstream, below. After that, in Section VII-B, we discuss Rootstock and merged mining, followed by Ethereum's Plasma in Section VII-C and Cardano sidechains in Section VII-D.

A. FEDERATED PEGS, BLOCKSTREAM'S ELEMENTS AND LIQUID

The idea of federated pegs was originally described by Back *et al.* in Appendix A of [14], and probably elsewhere before that, as part of the Bitcoin community folklore. A *pegged* sidechain is defined as a sidechain to which assets of a main chain can be transferred. However, since the Bitcoin blockchain lacks the scripting capabilities required to perform assets transfer, the authors proposed the idea of *federated pegs*, i.e., a fixed set of known and semi-trusted nodes, known as *functionaries*, that take care of moving the assets back from the sidechain to the main chain. The functionaries jointly agree to form a Byzantine consensus on some outcomes and indicate their agreement in the main chain by signing a k-out-of-n multi-signature (multi-sig) transaction. In the Bitcoin context, the functionaries may simply observe the Bitcoin chain and whenever they recognize there is an extension they know about, they enter their signature.

For implementing sidechains, the functionaries would observe both the main chain, verifying that the initial transaction freezing the assets is still valid and not spent, and the sidechain, looking for a valid transaction that freezes (or destroys) some sidechain assets while requesting them to be unfrozen in the main chain. Once they see both the transactions in the main chain and in the sidechain being valid, for a sufficiently long time, they add their signature in the main chain to unfreeze the assets there.

In the Bitcoin context, the federated pegs were first successfully implemented by the Elements Project[16] in late 2016. The Elements Project is a loose collection of various experimental activities on Bitcoin, utilizing sidechains. Elements is based on a ''strong federation'' consensus model [17], which relies on the collective actions of mutually-distrusting participants, the *Functionaries*, instead of a PoW consensus model [46]. The Functionaries include *Block Signers*, which participate in creating blocks through their signatures that are counted towards a threshold needed to validate proposed blocks, and *Watchmen*, which are responsible for moving assets in and out of the sidechain by signing multi-signature transactions.

Liquid[17] is an implementation of a sidechain based on the Elements framework. The goal of Liquid is to provide a permissioned blockchain with different features, capabilities, and benefits compared to the Bitcoin blockchain. In addition to faster trading, the benefits over Bitcoin include higher privacy.

[16] https://elementsproject.org
[17] https://blockstream.com/liquid

Other blockchain proposals, such as Stratis,[18] have recently supported sidechains based on a two-way federated peg model.

The work in [47] discusses an approach for constructing sidechains for a specific blockchain, Horizen,[19] but can be applied to other blockchains that implement the proposed cross-chain transfer protocol for transferring tokens (value) from the main chain to the sidechain. These sidechains may employ arbitrary consensus protocols, while the main chain (Horizen) remains agnostic with respect to individual sidechains. Transferring tokens from the main chain to the sidechain can be performed using a special transaction on the main chain that ''burns'' tokens and provides receipts that can be presented on the sidechain in order to create the corresponding amount of assets on the sidechain; Such an approach for transferring value from the main chain to the sidechain is similar to other proposals [14], [48]. Transfer of assets in the opposite direction, from the sidechains to the main chain, requires *Certifiers* to sign backward asset transfers. Nodes can register as Certifiers by locking some amount of their stake. The total amount of stake that is locked determines the amount of backward asset transfers.

B. MERGED MINING AND ROOTSTOCK

Merged mining allows performing Proof-of-Work (PoW) simultaneously for several DLTs that use the same underlying algorithm (e.g., SHA256 hash function). Merged mining can be applied to blockchains, including sidechains, whose block-header definition includes a part of Bitcoin's header [14]. This allows less popular DLTs with lower PoW rate to gain additional PoW power by relying on PoW performed for more popular distributed ledgers, improving their resilience (cf. [49]).

Sergio Lerner of RSK Labs[20] has attempted to analyze sidechains and related techniques in a working paper [48], mostly from an economic incentives point of view. Rootstock, which was initially proposed in 2014, is being released in stages, following its initial mainnet launch in early January 2018. Rootstock requires a relatively large change to Bitcoin and merged mining, enabling what they call *drivechains*. In drivechains, locking and unlocking of assets is controlled by the (merged) miners, while in federated pegs it is done by the functionaries outside the main chain. Drivechain[21] is another effort for creating multiple blockchains that are linked with a two-way peg to Bitcoin. It uses merged mining, similar to Rootstock, in addition to SPV proofs.

C. ETHEREUM'S PLASMA

Plasma [50] is a proposal for creating hierarchical trees of sidechains (or child blockchains) using a series of smart contracts. The Ethereum blockchain (root chain) needs to process only a small amount of commitments from sidechains, which however can perform a large amount of computations. Each sidechain is implemented through a smart contract, which can be governed by its own set of rules and constraints. This includes the ability to customize the permissioned set of nodes that participate in the production of blocks. Plasma sidechains use Proof-of-Stake consensus. Mining is done with full security only on the root chain. Unlike the Lightning and Raiden Networks, which work strictly for payments, Plasma extends the idea to Ethereum smart contracts.

Validators report the activity taking place on a child chain to the root chain, in the form of blockheader hashes rather than a full list of transactions performed on the child chain. Data is propagated only to parties that wish to validate the state of particular sidechains that they are interested in. The parties monitoring a particular sidechain are responsible for penalizing fraud. States within this child blockchain are enforced via fraud proofs, which are part of smart contract logic, that ensure that all state transitions are validated. Fraud proofs can also enforce an interactive protocol for fund withdrawals, similar to how HTLCs are used in the Lightning Network; these fund withdrawals need to be published on the root chain, hence require the same time that is necessary for performing transactions on the Ethereum blockchain.

Plasma is being actively developed and used by projects such as OmiseGo,[22] whose goal is to build a peer-to-peer cryptocurrency exchange platform, and Loom,[23] which provides an SDK environment for building distributed applications on their own sidechain, with a focus on large-scale online games and social-network applications.

D. CARDANO

Cardano's CSL (Cardano Settlement Layer) uses the sidechains concept for moving assets from the CSL to the CCL (Cardano Computation Layer) sidechain, or other blockchains that support the Cardano KMZ[24] protocol [52], for efficient Simplified Payment Verification (SPV) proofs. CSL is Cardano's main blockchain, which supports a very limited set of operations in order to achieve high security level. On the other hand CCL is a sidechain that supports more features, including experimental ones. The work in [51] proposes a procedure for constructing Non-Interactive Proofs of Proof-of-Work (NiPoPoWs). The non-interactive nature of the procedure refers to the fact that it involves a single messaging exchange, which is appropriate for transferring assets between two chains. Unlike a traditional blockchain client which must verify the entire linearly-growing chain of PoWs, clients based on NiPoPoWs can verify a certain blockchain property requiring resources only logarithmic in the length of the blockchain. NiPoPoWs solve two important

[18] https://academy.stratisplatform.com/Sidechains/sidechains-introduction.html

[19] https://www.horizen.global/

[20] https://www.rsk.co/

[21] http://www.drivechain.info/

[22] https://omisego.network

[23] https://loomx.io

[24] KMZ are the initials of Kiayias, Miller, and Zindros, who are the authors of the corresponding paper [51].

open questions for PoW-based consensus protocols: The problem of constructing efficient SPV clients and the problem of constructing efficient sidechain proofs.

VIII. LEDGER-OF-LEDGERS APPROACHES

In a ledger-of-ledgers approach, a new ''super-ledger'' is introduced, with the goal of having multiple sidechain-like ledgers or other subledgers, as shown in Figure 7. However, even though one of the overall goals of the interledger approaches is to move away from the ''one chain rules them all'' paradigm (an explicit motivation in the Polkadot whitepaper [53]), both Polkadot and Cosmos, the two main ledger-of-ledgers representatives, essentially introduce their own key ledgers that ''rule'' them all (in terms of interconnecting them).

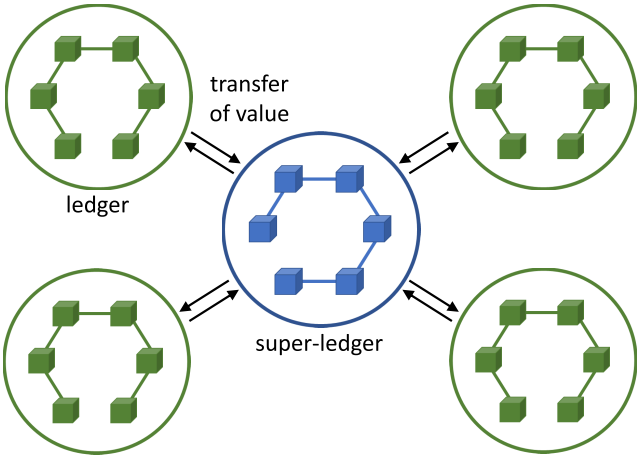


FIGURE 7. With the ledger-of-ledgers approaches a new super-ledger is introduced, which is the intermediary when two ledgers need to communicate.

With the interconnection ledger, similar to sidechain approaches, ledger-of-ledgers solutions can provide high scalability when interconnecting multiple sidechain-like ledgers, but at the cost of high complexity due to the presence of the interconnection ledger. Moreover, the transaction cost on the sidechains is smaller than the cost on the interconnection ledger.

Polkadot[25] is a proposal introduced by Gavin Wood, the author of the Ethereum yellow paper. Polkadot is open source, but most of the work appears to take place at Parity Technologies. While Polkadot is primarily described as a scalable heterogeneous ''multi-chain'', in reality it attempts to introduce a new, overarching relay-chain, upon which a large number of so-called *parachains* can be built; the relay-chain is responsible for finalizing all the transactions and can be used to interconnect permissionless and permissioned ledgers. Typically the parachains would be new types of blockchains using the Polkadot-specific Byzantine Fault Tolerance (BFT) consensus algorithm, inspired by Tangaora, Tendermint, and HoneyBadgerBFT. The BFT is further turned into a Proof-of-Stake (PoS) like system, periodically electing the set of validators randomly from a set of bonded potential validators using the size of their bonds (measured in Polkadot DOT tokens) as a measure of their stake. This, of course, depends on the bonds and consequently on the underlying Polkadot tokens having real-world value through an exchange or other mechanism. Polkadot itself provides no inherent application functionality other than allowing the parachains (including Ethereum and Bitcoin) to relay data and eventually value between them.

Polkadot can contain multiple parachains with differing characteristics in terms of privacy and trust. Scalability is achieved by spreading transactions across the chains, allowing them to process transactions in parallel. Parachains can be created by bonding DOT tokens. On the other hand, parachains can be removed by unbonding DOT tokens. This achieves a shared security model among the parachains, which enables individual chains to leverage the collective security of the whole system immediately when they are created.

Cosmos[26] is a project by the Interchain Foundation, a Swiss foundation registered in early 2017 by people from All In Bits, Inc. It is similar in structure to Polkadot. The Cosmos core is based on Tendermint, a BFT blockchain technology. The goal is to establish a heterogeneous network of PoS blockchains that can interoperate with one another. Developers can therefore build both public and private blockchains on top of the Tendermint core engine. Moreover, Cosmos explicitly aims at preserving the sovereignty of the sidechains.

The Cosmos Hub is the main blockchain that interconnects many other independent parallel blockchains, called zones [54]. The Cosmos hub and zones can implement a classical BFT consensus algorithm, such as Tendermint. The main token of the Cosmos Hub is the Atom, which is used for staking and governance of the blockchain. Unlike Polkadot, Cosmos does not require Atoms to be bonded to create a new zone. Each zone can have its own governance mechanism and policies. The Hub is responsible for ensuring the global invariance of the total amount of tokens held by individual users or by zones. From the Hub's perspective, a zone is a multi-asset dynamic-membership multi-signature account that can send and receive tokens using an Inter-blockchain Communication (IBC) protocol. Unlike Polkadot, zones submit transactions to hubs only for inter-blockchain transactions.

A special zone, called a *bridged zone*, is responsible for transactions between the Cosmos network and other blockchains such as Ethereum or Bitcoin. To achieve this, the bridged zone must follow the transaction on both the Cosmos network side and the interconnected blockchain. For the latter, a bridge account on the interconnected blockchain, e.g., Ethereum, can be used to send and receive ether tokens to other Ethereum accounts. When the ether tokens are received by the bridge contract, a corresponding account is created on

[25]https://polkadot.network

[26]https://cosmos.network/

TABLE 1. Comparison of interledger approaches.

Approach	Transfer or exchange of value	Interconnection trust mechanism	Complexity	Scalability	Transaction Cost	References
Atomic cross-chain transactions	Exchange of value	Hash and time-locks	Low	Medium (typically involve two chains)	Cost consists of transaction costs on both participating chains	[13], [21]–[27]
Transactions across a network	Exchange of value	Hash and time-locks	Medium (require routing of payments)	High (route payments via intermediate nodes)	Payment channels incur cost for opening and closing on-chain transaction; transaction cost can be very low if there are many intermediate off-chain transactions	[28]–[36]
ILP	Exchange of value	ILPv1: Hash and time-locks ILPv4: unconditional payment channels, legacy payment systems	Medium (requires routing of payments)	High (route payments via intermediate nodes)	Payment channels incur cost for opening and closing on-chain transaction; transaction cost can be very low if there are many intermediate off-chain transactions; focus is on open protocol for interconnection	[37]–[39]
Bridging	Exchange or transfer of value	Hash and time-locks (some proposals); Modules running on one or both of the interconnected chains	Medium-High	Medium (typically involve two chains)	Cost consists of transaction costs on both participating chains	[21], [40]–[45]
Sidechains	Transfer of value	Federated functionaries and multiparty signatures, SPV proofs, or validators with hash and time-locks	High	High (exploit transaction locality)	Transaction cost on sidechains is smaller than on the main chain	[14], [17], [46]–[51]
Ledger-of-ledgers	Transfer of value	Requires an additional interconnection ledger	High (require interconnection ledger)	High (exploit transaction locality)	Transaction cost on sidechains is smaller than on the interconnection chain	[18], [53], [54]

the bridge zone with the corresponding balance in ''bridged ether''. In the opposite direction, when sending ether from the bridge account to another Ethereum account, the corresponding amount of bridged ether is destroyed.

English *et al.* [18] described the Uberledger framework, a hierarchical meta-blockchain layer and an open-source initiative that aimed at preserving all information related to past transactions across blockchains, thereby allowing the behavior of the parties to accumulate, forming a basis for reputation and trust.

Ripple[27] has the goal of supporting global money transfers using blockchains. Ripple has been instrumental in developing the Interledger Protocol[28] (ILP) discussed in Section V, which, however, in its most recent incarnation, ILPv4, does not rely on DLTs for the payments (or asset transfers). However, Ripple has also developed XRP, Ripple's coin or ''digital asset for payments'' and its own blockchain, the Ripple Consensus Ledger (RCL), which serves as the root ledger for XRP, with decentralized governance and consensus mechanism with the goal of global payments and value exchange between any type of ledgers and for various types of assets. Thus, Ripple allows payments and asset exchanges across all types of ledgers, from traditional bank ledgers and fiat currencies to tokens on blockchains, cryptocurrencies in wallets, and arbitrary types of assets.[29] Therefore, it can be considered a ledger-of-ledgers technology, proclaiming corresponding properties, for instance faster and cheaper transactions than traditional public blockchains, such as Bitcoin and Ethereum, but with similar decentralized characteristics. However, Ripple expects that ILP will enhance the reach and impact of the RCL and that it will expand its role in global payments.

The aforementioned projects attempt to build yet another distributed ledger technology to allow a set of underlying blockchains (parachains, sidechains) to pass information and value between each other. Both Polkadot and Cosmos approach this by relying on Byzantine consensus, but changing that into a bonded one, thereby making it a PoS system. Hence, their security at the interledger level depends mainly on creating yet another ecosystem and token, with the tokens apparently having value due to them functioning as bonds or stakes in the ecosystem and thereby giving power to the ecosystem. Whether such a practice has sustainable social value should be evaluated very carefully.

Similar to bridging approaches, ledger-of-ledgers approaches, such as Polkadot and Cosmos, can provide a decentralized exchange that does not require the trading parties to be online, as is the case for direct peer-to-peer trading parties that use atomic cross-chain transactions. The decentralized exchange operates on its own blockchain (the super-ledger blockchain) and receives transaction requests from the parties. The transaction requests are executed by the blockchain nodes, thus providing a decentralized trust system for reliably executing transactions.

IX. DISCUSSION AND COMPARISON

Table 1 below summarizes the comparison of the interledger solutions presented in this survey.

[27] https://ripple.com
[28] https://ripple.com/insights/implementing-the-interledger-protocol/
[29] https://ripple.com/faq/

A. TRANSFER OR EXCHANGE OF VALUE

Hash-lock and time-lock mechanisms are used to exchange the value (tokens) between two parties on two blockchains. Hence, the atomic cross-chain transactions, transactions-across-a-network, ILP, and some bridging solutions that use hash and time-lock mechanisms implement the exchange of value, whereas all the other solutions implement the transfer of value, whereby value is locked or destroyed on one chain and unlocked or created on another.

A key difference is that the exchange of value is dependent on the exchange rate of the assets being traded, which is not the case when value is transferred. When value is transferred the corresponding assets are controlled by the new chain.

B. INTERCONNECTION TRUST MECHANISM

The interconnection trust mechanism defines where the immutable state of the interconnected transactions is recorded. With hash-lock and time-lock mechanisms, the state is recorded on the ledgers with the interconnected transactions. This is similar to bridging and sidechain approaches, where, however, some nodes (the verifiers) need to view and/or process the whole or a subset - in the case of Simplified Payment Verification (SPV) proofs - of the ledger data on the two interconnected chains. On the other hand, ledger-of-ledgers approaches require an interconnection ledger where the state related to the interconnection of different ledgers resides, making them more complex.

C. COMPLEXITY

Hash-locks and time-locks involve hashing and comparisons, which are inherently simple mechanisms. Also, using hash-locks and time-locks to cryptographically link transactions on different chains is achieved by storing the same hash value and relying on the same hash function, thereby requiring the same secret to unlock. Hence, the interledger solutions that use hash-lock and time-lock mechanisms have low complexity. However, depending on the consensus mechanism used in the interconnected ledgers that determines the transaction delay, which in turn can depend on whether the interconnected ledgers are public or permissioned, the timeouts defined in such solutions can be long, in the order of one or two days. Indeed, the transaction delay for PoW-based consensus is typically much longer than PoS, PoA, and BFT-based consensus mechanisms; the latter two mechanisms are typically used in permissioned ledgers. Hence, finalizing an interledger transaction, especially when public ledgers are involved, can be particularly slow.

Transactions across a network and ILP incur the routing cost to propagate transactions across a network of nodes. This places them at a higher computational cost compared to atomic cross-chain transactions, albeit achieving faster transaction finalization. However, routing costs tend to be orders of magnitude lower than the cost of downloading and processing an entire chain, in terms of the required computational resources, therefore we can deduce that these mechanisms are expected to be computationally lighter than bridging, sidechains, and ledger-of-ledgers approaches.

Bridging and sidechain approaches incur a higher degree of computational cost, as they can require a set of nodes (e.g., the functionaries) to view and process the entire interconnected chains. This operation is computationally significantly more costly than atomic cross-chain transactions, as functionaries are required to keep downloading and processing the new blocks of all interconnected chains. This, however, substantially speeds up transaction finalization time, as it renders the need for long time-lock timeouts irrelevant. A particularly severe facet of this mechanism's computational cost has to do with bootstrapping functionaries, as processing an entire chain is an inherently slow and resource-demanding operation (e.g., for Bitcoin it would amount to downloading and processing over 200 GB of data as of January 2019). Techniques such as SPV proofs (Section VII-D), allow the partial processing of data residing in the interconnected chains, with a compromise on security guarantees. Note that for bridging approaches that are based on hash and time-locks, the complexity cost can be lower than approaches that require viewing and processing the entire interconnected chains; nevertheless, such bridging approaches have a higher complexity compared to atomic cross-chain transactions with peer-to-peer interaction between two parties, since the former provide additional functionality, such as verification and discovery services. Ledger-of-ledgers approaches are also complex solutions, since they require the presence of another, interconnecting ledger.

D. SCALABILITY

Scalability is discussed in terms of the total number of transactions per unit of time that a solution can support. In this sense, a single ledger has low scalability. Atomic cross-chain transactions across two chains and bridging approaches, which typically involve two chains, provide higher scalability compared to a single ledger, since they can exploit transaction locality to reduce the number of transactions across the two chains. Sidechains and ledger-of-ledgers approaches involve multiple sidechains, hence can further exploit locality to reduce the transactions between the sidechains and the mainchain or the interconnecting chain, and in doing so achieve a higher degree of scalability compared to solutions focusing on the interconnection of two chains. Moreover, whereas most sidechain and ledger-of-ledgers approaches have a two-level hierarchy, some proposals, such as Ethereum's Plasma, go one step further and propose a hierarchy with multiple levels of sidechains.

Transactions across a network and ILP achieve high scalability in a way that is close to how scalability is achieved in the Internet: by using intermediate nodes to route transactions from an initiating party to a receiving party and distributing the transactions in the whole network across these intermediate nodes. The approach to increase scalability by distributing transactions through intermediate nodes and the

approach that considers a hierarchy of main (or interconnecting) chain and sidechains are fundamentally different in terms of the assumed and supported trust. Routing payments through intermediate nodes, following a series of interconnected micropayment channels that use hash and time lock mechanisms, does not assume trust between the intermediate nodes. Additionally, micropayment channels achieve scalability by using off-chain transactions, where transactions are performed in a peer-to-peer manner between parties, hence do not incur the transaction confirmation delay nor transaction cost of on-chain transactions. On the other hand, sidechains and ledger-of-ledgers solutions achieve scalability at the expense of a reduced level of trust for asset trading in the sidechains, compared to the level of trust on the main or interconnecting chain.

At this point, it is worth noting that significant work has been done aiming at improving blockchain scalability with respect to transaction throughput, mainly focusing on *sharding* techniques. The idea is to assign the validation of each transaction to a *shard* (i.e., a subset) of the total set of validators, rather than to the entire set. Sharding the validation load implies that each validator is responsible for only a fraction of submitted transactions, thus arbitrarily increasing validation throughput at the expense of security. Picking shards through a verifiably random distributed mechanism is vital to preventing malicious collusions. Omniledger [55] and RSCoin [56] present high-throughput sharding-based solutions for permissionless blockchains, while Channels [57] introduces a sharding mechanism for permissioned ones, implemented on Hyperledger Fabric. However, this vector of scalability, targeting intraledger transactions is orthogonal to the interledger operations we are exploring in this work.

E. TRANSACTION COST

Our final comparison criterion considers the transaction cost of the various solutions. For solutions involving the interconnection of two ledgers, which is the case with atomic cross-chain transactions and bridging, the cost is determined by the transaction cost on the two ledgers. Transactions across a network and ILP are forwarded along payment paths that are an interconnection of payment channels. Payment channels include only two on-chain transactions, for opening and closing the payment channel, while all intermediate transactions occur off-chain. Hence, the transaction cost for these solutions can be very low if there are many intermediate off-chain transactions. Note that most of the observations for ILP are the same as the transaction-across-a-network approaches, since ILP can use hash-lock and time-lock contracts, which are used in the transaction-across-a-network approaches, and payments are routed across a network of payment channels. The main difference is that ILP focuses on defining an open protocol for the interconnection. Intermediaries, such as connectors in ILP, can charge a fee for forwarding payments. With sidechains and ledger-of-ledgers approaches, the cost of transactions on sidechains is typically smaller than the cost of transactions on the main or interconnecting chain. The overall cost depends on the percentage of transactions that are conducted inside the sidechain and the transactions between the sidechain and the main chain. When the transaction locality is high, i.e., a large percentage of the transactions are conducted inside the sidechains, the overall cost will be lower.

X. CONCLUSION

The distributed ledgers community has acknowledged the gains that can be achieved from utilizing the interconnection of multiple ledgers with different features and advantages, while also supporting innovation which is critical to the field. Because there are no really established standards for combining several DLTs and the area is under continuous development, our focus in this survey has been to identify the repeating patterns that are shared among the various solutions proposed, which include atomic cross-chain transactions, sidechains, bridging approaches, transactions across a network of payment channels, ledger-of-ledgers approaches, and the Interledger Protocol (ILP).

The identification of the fundamental features of the various solutions allows us to compare the solutions according to different criteria, which will continue to be important with further developments in the area. The first is whether the solution supports the transfer or the exchange of value. With transfer, value moves from one ledger to another. On the other hand, with exchange the value (tokens) on different ledgers are moved from one account to another on the same ledger. Interledger solutions differ in where the immutable state of the interconnected transactions is recorded, which can be on the two (or more) ledgers that are interconnected or on an interconnection ledger. Related to complexity, hash-locks and time-locks that cryptographically link transactions on different chains are the simplest mechanisms. Sidechains and ledger-of-ledgers approaches are hierarchical systems with high scalability in terms of the number of transactions they can support. Transactions across a network and ILP achieve high scalability in a way that is close to how scalability is achieved in the Internet: by routing transactions across paths that are an interconnection of payment channels, whose transactions can be conducted off-chain. Finally, the various solutions differ in the cost of transactions, with sidechains having a lower transaction cost compared to public or main chains, and off-chain transactions that occur with peer-to-peer exchange having zero transaction cost.

Preliminary interledger approaches for value transfers across blockchains have already been in use for the last three years. For instance, BTC Relay, released in early 2016, allows Bitcoin payments for executing Ethereum smart contracts.

Currently, various IoT applications are being envisaged and prototyped to allow decentralized IoT-based supply chains to guarantee various properties and to support transaction non-repudiation. One such application using blockchains to secure a ''field-to-fork'' food-chain is being prototyped by project SOFIE[30] (Secure Open Federation for Internet

[30]https://www.sofie-iot.eu/

Everywhere) [58]. The information recorded in blockchains includes food production and storage data, acquired through diverse types of sensors. Various aspects of the process are controlled through actuation to guarantee safety and quality. However, this supply chain is long, with many independent actors involved, from farmers and farm associations, through transportation companies and independent transporters, storage locations, to super-markets and customers, and customer and food-safety associations. These actors will be hard to agree on a single ledger, in particular since different ledgers excel at different use cases. Interledger approaches are, thus, very promising.

REFERENCES

[1] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST.IR.8202, Oct. 2018. [Online]. Available: https://doi.org/10.6028/NIST.IR.8202
[2] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018. doi: 10.1504/IJWGS.2018.10016848.
[3] D. K. Hays and M. J. Valek, "Crypto research report, edition III," Incrementum AG, Schaan, Liechtenstein, Tech. Rep., Jun. 2018. [Online]. Available: https://cryptoresearch.report/wp-content/uploads/2018/06/Crypto-Research-Report-June-2018-3.pdf
[4] P. Tasca and C. J. Tessone, "Taxonomy of blockchain technologies. Principles of identification and classification," *CoRR*, May 2017. [Online]. Available: https://arxiv.org/abs/1708.04872
[5] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2017, pp. 557–564.
[6] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols," in *Proc. IACR Cryptol. ePrint Arch.*, 2015, p. 1019.
[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[8] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, "Decentralized consensus for edge-centric Internet of Things: A review, taxonomy, and research issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2018. doi: 10.1109/ACCESS.2017.2779263.
[9] 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 Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.
[10] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 243–252.
[11] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.
[12] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, to be published. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17318332. doi: 10.1016/j.future.2017.08.020.
[13] M. Herlihy, "Atomic cross-chain swaps," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2018, pp. 245–254.
[14] 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," Blockstream, Victoria, BC, Canada, Tech. Rep., 2014. Accessed: Jul. 9, 2019. [Online]. Available: http://www.blockstream.com/sidechains.pdf
[15] Z.-D. Chen, Z. Yu, Z.-B. Duan, and K. Hu, "Inter-blockchain communication," in *Proc. Int. Conf. Comput. Sci. Technol. (CST)*, Dec. 2017, pp. 1–7. [Online]. Available: http://dpi-proceedings.com/index.php/dtcse/article/download/12539/12074
[16] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography Data Security*. Berlin, Germany: Springer, 2016, pp. 106–125.
[17] J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, and M. Friedenbach, "Strong federations: An interoperable blockchain solution to centralized third-party risks," *CoRR*, Dec. 2016. [Online]. Available: https://arxiv.org/abs/1612.05491
[18] S. M. English, F. Orlandi, and S. Auer, "Disintermediation of inter-blockchain transactions," *CoRR*, Sep. 2016. [Online]. Available: https://arxiv.org/abs/1609.02598
[19] H. Sun, H. Mao, X. Bai, Z. Chen, K. Hu, and W. Yu, "Multi-blockchain model for central bank digital currency," in *Proc. Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Dec. 2017, pp. 360–367.
[20] M. Tate, D. Johnstone, and E. Fielt, "Ethical issues around crowdwork: How can blockchain technology help?" in *Proc. Australas. Conf. Inf. Syst. (ACIS)*, 2017. [Online]. Available: https://eprints.qut.edu.au/115042/
[21] V. Buterin, "Chain interoperability," R3, R3 Rep., Sep. 2016. Accessed: Jul. 9, 2019. [Online]. Available: https://www.r3.com/reports/chain-interoperability/
[22] T. Nolan. *Alt Chains and Atomic Transfers*. Accessed: Jul. 9, 2019. [Online]. Available: https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949
[23] *Atomic Cross-Chain Trading*. Accessed: Jul. 9, 2019. [Online]. Available: https://en.bitcoinwiki.org/wiki/Atomic_cross-chain_trading
[24] *Hashed Time-Lock Contracts (HTLC)*. Accessed: Jul. 9, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Hashed_Timelock_Contracts
[25] *Hashlock*. Accessed: Jul. 9, 2019. [Online]. Available: https://en.m.bitcoinwiki.org/wiki/Hashlock
[26] *Timelock*. Accessed: Jul. 9, 2019. [Online]. Available: https://en.bitcoinwiki.org/wiki/Timelock
[27] *Atomic Swap*. Accessed: Jul. 9, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Atomic_swap
[28] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Proc. 17th Int. Symp. Self-Stabilizing Syst.*, Nov. 2015, pp. 3–18.
[29] J. Poon and T. Dryja. (Jan. 2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. Accessed: Jul. 9, 2019. [Online]. Available: https://lightning.network/lightning-network-paper.pdf
[30] J. Hosp, T. Hoenisch, and P. Kittiwongsunthorn, "COMIT—Cryptographically-secure off-chain multi-asset instant transaction network," *CoRR*, Oct. 2018. [Online]. Available: https://arxiv.org/abs/1810.02174
[31] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, "Flare: An approach to routing in lightning network," Bitfury Group, Tech. Rep., Jul. 2016. Accessed: Jul. 9, 2019. [Online]. Available: http://bitfury.com/content/5-white-papers-research/whitepaper_flare_an_approach_to_routing_in_lightning_network_7_7_2016.pdf
[32] P. Wang, H. Xu, X. Jin, and T. Wang, "Flash: Efficient dynamic routing for offchain networks," *CoRR*, Feb. 2019. [Online]. Available: https://arxiv.org/abs/1902.05260
[33] B. Vu. (May 30, 2018). *Exploring Lightning Network Routing*. Accessed: Jul. 9, 2019. [Online]. Available: https://blog.lightning.engineering/posts/2018/05/30/routing.html
[34] G. Malavolta, "Anonymous multi-hop locks for blockchain scalability and interoperability," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, Feb. 2019, pp. 1–30.
[35] C. Burchert, C. Decker, and R. Wattenhofer, "Scalable funding of bitcoin micropayment channel networks," in *Proc. 19th Int. Symp. Stabilization, Saf., Secur. Distrib. Syst. (SSS)*, Nov. 2017, pp. 361–377.
[36] G. D. Stasi, S. Avallone, R. Canonico, and G. Ventre, "Routing payments on the Lightning Network," in *Proc. IEEE Int. Conf. Blockchain*, Aug. 2018.
[37] S. Thomas and E. Schwartz. *A Protocol for Interledger Payments*. Accessed: Jul. 9, 2019. [Online]. Available: https://interledger.org/interledger.pdf
[38] A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," in *Proc. 25th Int. Conf. Companion World Wide Web*, Apr. 2016, pp. 281–282.
[39] *Interledger Protocol V4*. Accessed: Jul. 9, 2019. [Online]. Available: https://interledger.org/rfcs/0027-interledger-protocol-4
[40] A. Culwick and D. Metcalf. *The Blocknet Design Specification*. Accessed: Jul. 9, 2019. [Online]. Available: https://www.blocknet.co/wp-content/uploads/2018/04/whitepaper.pdf
[41] ARK. (Apr. 1, 2019). *ARK Ecosystem Whitepaper. Version 2.0.0*. Accessed: Jul. 9, 2019. [Online]. Available: https://ark.io/Whitepaper.pdf
[42] Wanchain. (2017). *Building Super Financial Markets for the New Digital Economy. White Paper Version 0.9.0*. Accessed: Jul. 9, 2019. [Online]. Available: https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf
[43] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Secur., Adv. Cryptol. (ASIACRYPT)*, 2001, pp. 552–565.
[44] Aion. (2017). *Aion White Paper Version 1.0.0*. Accessed: Jul. 9, 2019. [Online]. Available: https://aion.network/media/en-aion-network-technical-introduction.pdf

[45] S. Hejazi-Sepehr, R. Kitsis, and A. Sharif. (Jan. 2019). *Transwarp-Conduit: Interoperable Blockchain Application Framework*. Accessed: Jul. 9, 2019. [Online]. Available: https://aion.network/media/TWC_Paper_Final.pdf
[46] *How Elements Works and the Roles of Network Participants*. Accessed: Jul. 9, 2019. [Online]. Available: https://elementsproject.org/how-it-works
[47] A. Garoffolo and R. Viglione. (Oct. 2018). *Sidechains: Decoupled Consensus Between Chains*. Accessed: Jul. 9, 2019. [Online]. Available: https://www.horizen.global/assets/files/Horizen-Sidechains-Decoupled-Consensus-Between-Chains.pdf
[48] S. D. Lerner. (Apr. 2016). *Drivechains, Sidechains, and Hybrid 2-Way Peg Designs*. [Online]. Available: https://docs.rsk.co/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf
[49] A. Judmayer, A. Zamyatin, N. Stifter, A. Voyiatzis, and E. Weippl, "Merged mining: Curse or cure?" in *Proc. Int. Workshop Cryptocurrencies Blockchain Technol. (CBT)*, Sep. 2017, pp. 316–333.
[50] J. Poon and V. Buterin. (Aug. 2017). *Plasma: Scalable Autonomous Smart Contracts*. [Online]. Available: https://plasma.io
[51] A. Kiayias, A. Miller, and D. Zindros, "Non-interactive proofs of proof-of-work," *IACR Cryptol. ePrint Arch.*, 2017, pp. 1–42.
[52] *Cardano Sidechains*. Accessed: Jul. 9, 2019. [Online]. Available: https://whycardano.com/#sidechains
[53] L. Wood. (Nov. 2016). *Polkadot: Vision for a Heterogenous Multi-Chain Framework*. Accessed: Jul. 9, 2019. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf
[54] *Cosmos Whitepaper*. Accessed: Jul. 9, 2019. [Online]. Available: https://cosmos.network/resources/whitepaper
[55] E. Kokoris-Kogias, P. S. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. A. Ford. (2018). *Omniledger: A Secure, Scale-Out, Decentralized Ledger Via Sharding*. [Online]. Available: [Online]. Available: http://infoscience.epfl.ch/record/255586
[56] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," 2015, *arXiv:1505.06895*. [Online]. Available: https://arxiv.org/abs/1505.06895
[57] E. Androulaki, C. Cachin, A. D. Caro, and E. Kokoris-Kogias, "Channels: Horizontal scaling and confidentiality on permissioned blockchains," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*, 2018, pp. 111–131.
[58] A. Karila, Y. Kortesniemi, D. Lagutin, P. Nikander, S. Paavolainen, N. Fotiou, G. C. Polyzos, V. A. Siris, and T. Zahariadis, "Secure open federation for Internet everywhere," in *Proc. Workshop Decentralized IoT Secur. Standards (DISS) Conjunct Netw. Distrib. Syst. Secur. Symp. (NDSS)*, Feb. 2018.

VASILIOS A. SIRIS (S'95–M'98) received the Diploma degree in physics from the National and Kapodistrian University of Athens, in 1990, the M.S. degree in computer science from Northeastern University, Boston, USA, in 1992, and the Ph.D. in computer science from the University of Crete, in 1998.

From 2002 to 2009, he was an Assistant Professor with the Department of Computer Science, University of Crete, and from 1993 to 2011, he was a Research Associate with the Institute of Computer Science, Foundation for Research and Technology—Hellas (FORTH). He was a Visiting Researcher with the Statistical Laboratory, University of Cambridge, and the Research Laboratory of British Telecommunications (BT), U.K. He has been with the Department of Informatics, Athens University of Economics and Business, since 2009, where he is currently an Associate Professor and the Director of the Computer and Communications Systems Division. His current research interests include trusted communication in the Internet of Things (IoT), the application of blockchain and distributed ledger technology to the IoT, resource management and traffic control in wired and wireless networks, and the architecture of future mobile and pervasive communication systems. He is an Area Editor of the *Computer Communications Journal* and has one international and two Greek patents in the area of resource management in wireless and mobile networks.

PEKKA NIKANDER received the M.Sc. and Ph.D. degrees (Hons.) in computer science from Helsinki University Technology [(HUT), a predecessor of Aalto University], Finland, in 1992 and 1999, respectively.

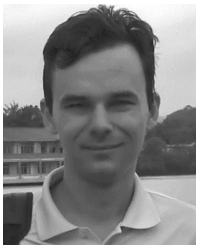
In 1988, he founded Nixu, a leading Nordic security company consultancy listed at Nasdaq OMX. From 1998 to 2011, he was with Ericsson Research, serving in various internal and external roles, including acting as an Associate Professor with HUT and being a member of the Internet Architecture Board, from 2005 to 2006. From 2011 to 2016, he co-founded four startup companies, including Solu Machines, and acted as the Chief Software Architect in two others, Senseg and PulseOn. Since 2017, he has been a Professor of industrial internet with Aalto University, Finland. His research interest includes the effect of the anti-rival properties of data and information to the structure of the economy.

SPYROS VOULGARIS received the Diploma degree in computer science from the University of Patras, Greece, in 1997, the M.Sc. degree from the University of Michigan, USA, in 1999, and the Ph.D. degree in computer science from Vrije Universiteit Amsterdam, in 2006. He was a tenured Assistant Professor with the Department of Computer Science, Vrije Universiteit Amsterdam. He joined the Athens University of Economics and Business as an Assistant Professor, in 2018.

He has served as a Postdoctoral Researcher with ETH Zürich, Switzerland, from 2006 to 2008. He has paid a number of research visits, including Microsoft Research Cambridge, INRIA Rennes, and the University of Bologna, while he has worked for HP Labs, Palo Alto, CA, USA, and for Hughes Network Systems, Germantown, MD, USA. His research interests include Internet-scale distributed systems, distributed ledger technology (blockchains), publish/subscribe protocols, big data infrastructures, information dissemination, peer-to-peer and epidemic algorithms, large-scale self-organization, mobile ad-hoc networks, and sensor networks.

NIKOS FOTIOU received the Diploma degree in information and communication systems engineering from the University of the Aegean, Samos, Greece, in 2005, the M.Sc. degree in internetworking from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2007, and the Ph.D. degree in computer science from the Athens University of Economics and Business (AUEB), Athens, Greece, in 2014.

Since 2014, he has been a Researcher with the Mobile Multimedia Laboratory, AUEB. His research interests include future Internet architectures, security and privacy, the IoT systems, and applications of the distributed ledgers technology.



DMITRIJ LAGUTIN received the M.Sc. (Tech.) degree from the Helsinki University of Technology, Finland, in 2005, and the D.Sc. (Tech.) degree from Aalto University, Finland, in 2010.

He was a Researcher in several research projects with the Helsinki University of Technology and Aalto University, including EU FP7 PSIRP and PURSUIT an EU H2020 POINT projects. He is currently a Coordinator and Research Fellow with the EU Horizon 2020 SOFIE Project at Aalto University, Finland. His research interests include network security and privacy, the Internet of Things, blockchains, and future network technologies.

GEORGE C. POLYZOS (S'79–M'82) received the Diploma degree in electrical engineering from National Technical University, Athens, Greece, and the M.A.Sc. degree in electrical engineering and the Ph.D. degree in computer science from the University of Toronto, Canada.

He was a Professor of computer science and engineering with the University of California at San Diego, San Diego, where he was the Codirector of the Computer Systems Laboratory, a member of the Steering Committee of the Center for Wireless Communications, and a Senior Fellow of the San Diego Supercomputer Center. Under his leadership the MMlab has participated in a series of research projects funded by the EC (PSIRP, PURSUIT, and POINT), the ESA (ϕSAT), and Greece (I-CAN) that codeveloped Publish-Subscribe Internetworking, an Information-Centric Networking architecture. PURSUIT received the Future Internet Award, in 2013. He is leading the Mobile Multimedia Laboratory (MMlab), Athens University of Economics and Business, where he is currently a Professor of computer science and the Director of the M.Sc. Program in computer science. His current research interests include the Internet-of-Things and ubiquitous computing, security and privacy, Internet architecture and protocols, information-centric networking, mobile multimedia communications, wireless networks, and systems performance evaluation.

Dr. Polyzos has served on journal editorial boards, as a Special Issue Guest Editor, and on committees of many conferences and workshops. He has chaired the Steering Committee of the ACM SIGCOMM conference on Information-Centric Networking. He is currently on the Steering Committee of the Wireless and Mobile Networking Conference (IFIP TC6 WG 6.8) and an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING.