Do You Need a Distributed Ledger Technology Interoperability Solution?

RAFAEL BELCHIOR, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal and Quant Network, United Kingdom
LUKE RILEY, Quant Network, United Kingdom
THOMAS HARDJONO, Massachusetts Institute of Technology, United States
ANDRÉ VASCONCELOS, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
MIGUEL CORREIA, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Entrepreneurs, enterprises, and governments are using distributed ledger technology (DLT) as a component of complex information systems, and therefore interoperability capabilities are required. Interoperating DLTs enables network effects, synergies and, similarly to the rise of the Internet, it unlocks the full potential of the technology. However, due to the novelty of the area, interoperability mechanisms (IM)[1] are still not well understood, as interoperability is studied in silos. Consequently, choosing the proper IM for a use case is challenging.

Our paper has three contributions: first, we systematically study the research area of DLT interoperability by dissecting and analyzing previous work. We study the logical separation of interoperability layers, how a DLT can connect to others (connection mode), the object of interoperation (interoperation mode), and propose a new categorization for IMs. Second, we propose the first interoperability assessment for DLTs that systematically evaluates the interoperability degree of an IM. This framework allows comparing the potentiality, compatibility, and performance among solutions.

Finally, we propose two decision models to assist in choosing an IM, considering different requirements. The first decision model assists in choosing the infrastructure of an IM, while the second decision model assists in choosing its functionality.

CCS Concepts: • **Computer systems organization → Dependable and fault-tolerant systems and networks**.

Additional Key Words and Phrases: survey, blockchain interoperability, standards, interconnected DLT networks, cross-chain transactions, cross-blockchain communication, DLT infrastructure, interoperability assessment framework

ACM Reference Format:
Rafael Belchior, Luke Riley, Thomas Hardjono, André Vasconcelos, and Miguel Correia. . Do You Need a Distributed Ledger Technology Interoperability Solution?. , ( ), 39 pages.

[1] we use interoperation mechanisms interchangeably with DLT interoperability solutions.

Authors' addresses: Rafael Belchior, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol, 9, Lisboa, Portugal, 1000-029 and Quant Network, 20-22 Wenlock Road, London, United Kingdom, N1-7GU, rafael.belchior@tecnico.ulisboa.pt; Luke Riley, Quant Network, 20-22 Wenlock Road, London, United Kingdom, N1-7GU, luke.riley@quant.network; Thomas Hardjono, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, United States, MA 02139, hardjono@mit.edu; André Vasconcelos, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol, 9, Lisboa, Portugal, 1000-029, andre.vasconcelos@tecnico.ulisboa.pt; Miguel Correia, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol, 9, Lisboa, Portugal, 1000-029, miguel.p.correia@tecnico.ulisboa.pt.

1 INTRODUCTION

Before a technology unlocks its full range of applications, it first undergoes underestimation. Distributed Ledger Technology (DLT), including blockchain, is no exception and is here to stay. A DLT (distributed ledger) or blockchain[2] implements a ledger that is shared across a network of nodes. Each node can create, broadcast, and validate *transactions*, which modify the distributed ledgers' state. DLTs typically provide support to run *smart contracts*, computer programs whose output is recorded on the ledger state. Smart contracts are triggered by transactions, which are recorded on the ledger. Nodes agree on the validity and ordering of transactions via a *consensus mechanism*. Typically, this environment provides transparency, tamper-resistance, and auditability of ledger information, providing desirable features that can alleviate some of the problems of the "centralized world".

Dozens of distributed ledger technology and blockchain systems [51] give rise to hundreds of blockchains [31] that, in its turn, support thousands of cryptocurrencies [31]. In the second quarter of 2021, decentralized exchanges (also called *automated market makers*) alone recorded a volume of $343 billion [34]. Along with Coinbase's total trading volume of $335 billion, the trends towards using blockchain for finance are increasing.

Payment networks, central bank digital currencies (CBDCs) and decentralized finance (DeFi) applications are already being leveraged by multiple players, such as (centralized and decentralized) hedge funds [130]. El Salvador adopted Bitcoin as a legal tender in June 2021. Several dozen projects on central bank digital currencies, including the Digital Pound consortium and the European Central Bank's Digital Euro [92] are displaying the increasing need for digitizing money [60]. Adoption seems inevitable as the world's financial ecosystems evolve [85]. Research suggests that the market for applications using DLTs will grow, with many organizations stating that blockchain is a critical priority [63, 103, 132], due to, for example, cost reduction. A recent report from Gartner predicts that "by 2023, 35% of enterprise blockchain applications will integrate with decentralized applications and services" [86]. Many blockchain ecosystems invest and promote projects that advance knowledge in cryptocurrencies [21, 58] and blockchain open-source research [69], bringing more adoption to the space.

Thus, blockchain is slowly but steadily becoming an infrastructure for global value exchange and distributed computation [78]. However, blockchains have been created as standalone networks, as autonomy from most external systems was sufficient for the first applications.

Moreover, the need to securely and seamlessly connect DLTs (integration) is still an open problem [12, 131, 133]. Connecting those blockchains and making them cooperate (i.e., achieving interoperability [30]) have a practical utility and importance [12, 133, 134]. It allows communication between systems to exchange data and assets (fungible and non-fungible), leading to a higher heterogeneity of solutions in the market, synergies between projects, and higher liquidity to end-users. This way, no blockchain should become a single point of failure. Digital identity, supply chain, healthcare, voting [40] and central bank digital currencies (CBDCs) [12, 130] are just a few use cases benefiting from a multiple blockchain approach. We believe that blockchain will be adopted *en masse* when blockchains can use the capabilities of other systems in a unified approach [30].

[2]We use the two terms interchangeably to mean a system with the characteristics explained in the rest of the paragraph. As a data structure, a blockchain is a distributed ledger but the opposite is not true. The reader is assumed to understand blockchain basics. For some references, please refer to [35, 85].

Key Takeway 1. *Integrating blockchains*

Several blockchain projects already have embedded integration capabilities with other projects (e.g., Hyperledger Fabric can execute Solidity smart contracts). However, these existing integrations do not imply interoperability, as reutilizing functionality of a system does not imply cooperation across systems.

To connect DLTs and centralized systems, one needs blockchain interoperability techniques. A *centralized system* can still be distributed, typically for scaling purposes, but its components are trusted and operate under the umbrella of one authority. More concretely, it is a system where the state consensus is decided by a single party or multiple parties under the same authority. A *decentralized system* is distributed system where various parties control different components of the distributed system, and no party is fully trusted by all. In our context, we can consider a decentralized system to be a system where the state consensus is decided by conflicting or competing multiple parties, where accountability (from an external viewer's point of view) of individual decisions is *assured*. Each party composing the system can vote autonomously and has different incentives from other parties.

Interoperability allows a set of systems to cooperate, to achieve a common goal [67] – it is "the ability of two or more systems to cooperate despite differences in language, interface, and execution platforms" [128]. Studied since the 1980s [62, 67, 87, 123, 128] , interoperability plays a major role connecting information systems. In this paper, we propose a framework to assess the maturity of a DLT-based application to adapt to other systems (potentiality), its interoperation capabilities (compatibility), and its performance.

More recently, over a dozen academic papers surveyed the state of blockchain interoperability, identifying a few dozen solutions (for an updated list, see page 10 of [12]). In those surveys, examples of interoperation between networks of the same and different technologies are studied. Findings show that integrating multiple blockchains allows enterprise systems to be connected to DLTs and enables the creation of multi-ledger decentralized applications. Those applications can ideally run arbitrary cross-chain logic across DLTs. Cross-chain logic (or cross-chain rules) can be executed against a pair of homogeneous DLTs (a pair of DLTs running the same DLT protocol) or heterogeneous DLTs (a pair of DLTs running different DLT protocols). Interoperating heterogeneous blockchains is complex, as there may be differences in the underlying cryptographic primitives, data models, consensus models, privacy assumptions, integration capabilities, and others.

Key Takeway 2. *Emergent Solutions*

General-purpose blockchain interoperability solutions are still relatively unexplored, where complex logic can be programmed across chains.

Despite recent evolutions connecting homogeneous blockchains, many unsolved challenges in blockchain interoperability theory and practice are exacerbated by the lack of standardization among APIs, data models, and processes. Thus, integrating with different DLTs is an error-prone and tedious task [47]. This is one of the reasons why it is still difficult for centralized systems to exchange assets with blockchains, despite advances in developing higher-level APIs that simplify this process [11, 118, 124]. Exchanging assets between blockchains comes with critical challenges, where we highlight security: cross-chain protocol security flaws have already resulted in the loss of hundreds of millions of dollars [55] in 2021 alone. Given a set of security, scalability, and decentralization requirements, choosing the right blockchain interoperability solution can help prevent attacks, diminish costs, and bring products to the market faster.

This work proposes to support the choice of an *interoperability mechanism* (IM), also known as interoperability solution. In specific, we support the choice of the infrastructure and functionality the IM, adjusted to specific project needs.

> Key Takeway 3. *Interoperability is not binary*
>
> To provide a better support for enterprise collaboration, synergies, and a richer ecosystem, integration processes should be verified and improved [36, 71]. Thus, integration is not a final step, but rather a continuous process that is also subject to change. Interoperability assessment tools exist for one to positioning systems in terms of how interoperable it is.

Research Questions and Contributions

Next, we present four fundamental interoperability questions that guide our research in this paper. Those are:

Research Question 1. *What is the status quo of DLT interoperability?*

Recent years have seen extensive work on IMs. Several surveys condensed that knowledge, focusing on public connectors (connecting public blockchains) [16, 24, 77, 115, 119, 137], architecture for blockchains [126], and others [13, 73, 74, 111, 120]. Some surveys provide a systematic overview of the area, showcasing more modern interoperability solutions [12, 88].

However, the classification of solutions is typically inconsistent across surveys, making it challenging for researchers to consistently evaluate available options. What is a consistent classification framework that can improve past work and improve understanding when classifying solutions? Furthermore, we aim to clarify theoretical contributions to the DLT interoperability research area, including the current capabilities, components, connection modes, interoperation modes, practical applications, limitations, and strong points. This guide can prove helpful to researchers and practitioners by providing a mental model of existing interoperability solutions.

Contribution: a unified conceptual model and classification framework for blockchain interoperability solutions.

Research Question 2. *How to assess the interoperability capabilities of an IM?*

Not all IMs provide the same interoperation capabilities. To measure it, one needs to consider several key questions. Measuring the maturity of a system to adapt to others requires asking *can the system interoperate with other systems as is?*, and *is the system able to be changed to adapt to other systems?*. Assessing interoperability between systems can be done by asking *how well can a pair of systems interoperate?*, and *what are the current problems or barriers that prevent the systems from interoperating better?*. Finally, measuring performance requires studying cross-chain latency, cross-chain throughput, and cross-chain costs associated with an IM.

Contribution: a framework to assess the interoperability capabilities of a system utilizing multiple DLTs (in terms of potentiality, compatibility, and performance), based on conceptual models [45, 71].

Research Question 3. *How to choose an IM?*

This research question concerns a problem posed by academics and practitioners alike: does my project need an IM solution? What is the most suited IM given specific requirements? We build on top of the proposed classification and interoperability assessment to answer these questions, presenting a framework for choosing an appropriate blockchain interoperability solution, both from the infrastructure and the functionality perspectives.

Contribution: a framework that allows one to choose a blockchain interoperability solution, considering a set of criteria defined by our blockchain interoperability model.

Structure of the Paper

In Section 2 we introduce the necessary background to read and understand this paper, including background on DLT interoperability, and examples that motivate DLT interoperability research. After that, we present the current state of DLT interoperability, including its several layers and components, and our model in Section 3. In the same section, we present the interoperation modes, the connection modes, and the (IM) solution categories. Section 4 presents our framework for assessing the interoperability of a DLT-based solution. After that, we present a decision model for choosing an IM for a DLT project based on the infrastructure and functionality of the IM, and two concrete examples. Section 5 presents the related work and future research challenges. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

In this section, we present the background and motivating examples.

2.1 Blockchain and Interoperability

Distributed Ledger Technologies, such as Hyperledger Fabric, Corda, and Ethereum implement DLT protocols. Each DLT protocol is defined by its protocol version, e.g., Hyperledger Fabric v2.3 [5], Corda v4 [20] or Ethereum London Hard Fork [23]. These technologies and version combinations describe how each DLT state can be updated via transactions and the specific protocol that the nodes follow to come to agreement on the DLT state, as Figure 1a illustrates.

DLT Networks and subnetworks. *DLT protocols* can be instantiated in *DLT networks*. DLT networks are groups of DLT nodes that make up a DLT system [72]. For instance, Hyperledger Fabric might be instantiated in a DLT network composed by an enterprise consortium. The Ethereum mainnet, Bitcoin mainnet, and Substrate-based networks, such as Polkadot, Kusama, and Rococo [105] are other examples. DLT networks are typically called Layer-1 DLTs.

Each DLT network can be partitioned into *subnetworks*. Nodes of a subnetwork contain logically separated state compared to another subnetwork [72].

Each subnetwork may offer different functionalities (e.g., data isolation, processing capabilities, governance) and security properties (e.g., partial consistency vs. consistency, better confidentiality, and so on). At least one node of each subnetwork must connect to another node of another subnetwork for these two-subnetworks to be contained within the same DLT network.

In Hyperledger Fabric, a subnetwork corresponds to a *channel*. Channels isolate execution environments and data from other channels belonging to the same Fabric network. Polkadot's Parachains could be considered subnetworks of the Polkadot network.

A *DLT network can therefore have multiple subnetworks*. If the DLT network state can not be divided into multiple subnetworks, for the sake of simplicity of our evaluation, we say this DLT network has one subnetwork. Any node in a DLT subnetwork is also a node of the DLT network, implying that DLT network nodes and DLT subnetwork nodes must be running the same DLT protocol. In *permissionless* DLT networks, every compatible DLT node can join the network. In contrast, in *permissioned* DLT networks, only compatible DLT nodes with permissions can join the network, where each DLT node may be assigned a particular role restricting the functions it can perform. There are two subcategories of permissioned DLT networks: *private permissioned* DLT networks, where DLT nodes do not provide public access to the data contained in the distributed ledger; and *public permissioned* DLT networks, where DLT nodes do provide public access to the data, such as via block explorers.

DLT Protocol	DLT Network	DLT subnetwork
Hyperledger Fabric 2.3	Carbon Emission Network (Hyperledger Fabric)	Carbon emission channel
Hyperledger Fabric 2.3	Carbon Emission Network (Hyperledger Fabric)	Travel channel
Ethereum 1.0 (geth version Faryar v1.10.15)	Ethereum Mainnet	Ethereum Mainnet
Ethereum 1.0 (Besu 21.10.6)	Ethereum Mainnet	Ethereum Mainnet
Ethereum 1.0 (geth version Faryar v1.10.15)	Ethereum Testnet Ropsten	Ethereum Testnet Ropsten
Ethereum 1.0 Private Network (Besu 21.10.6)	Private Ethereum Network	Privacy group 1
Ethereum 1.0 Private Network (Besu 21.10.6)	Private Ethereum Network	Privacy group 2
Polkadot 0.9.14	Polkadot	Relay chain
Polkadot 0.9.14	Polkadot	Parachain 1

Table 1. Examples of DLT networks, and its respective DLT protocol and subnetworks.

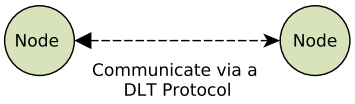
In permissionless DLT, anyone can run a node that interacts with the network; if it is permissioned, only nodes with permissions can access the network. Within a network boundary, a DLT stores the ledger state (which could be organised as a key-value store, such as the world state in Fabric [5], via the account model such as in Ethereum [23] or via a UTXO model such as in Bitcoin [97]), and have an identity management mechanism. Identities are then mapped to permissions that encode what each node can do on the network (in terms of reads and writes). Each identity typically has two main keys (public and private keys), an address, and a low-level storage. Nodes can perform updates to the ledger via *transactions*.

Transactions are "the smallest unit of a work process related to interactions with distributed ledgers" [72], that, parametrized and signed by its creator, can be issued against a smart contract, via a DLT node. Generally speaking (as different DLT technologies have different transaction lifecycles), transactions are sent to other DLT nodes via a network propagation protocol, such as a Gossip [35].

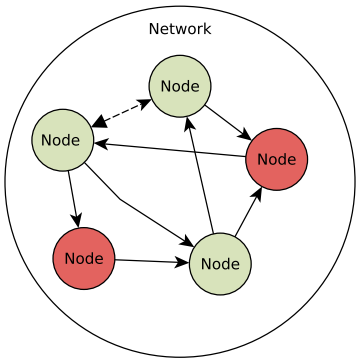
Internal Mechanisms. DLT nodes in the same DLT network will "gossip" to each other various messages, such as transactions to update the distributed ledger. DLT nodes agree on the order of transactions (and its content) to update the distributed ledger, by following a set of rules and procedures defined in a *consensus mechanism*. Typically, DLT networks have an *anti-sybil* component so that individual DLT nodes cannot replicate themselves to unfairly increase their influence on how the entire network reaches consensus. Some DLTs may allow for selectable consensus mechanisms (usually selectable only upon the genesis of the DLT network), such as with Ethereum or DLTs created with the Substrate framework. In contrast, other DLTs like Bitcoin have a hardcoded consensus mechanism.

How transactions affecting the distributed ledger are ordered gives different DLT types. A *blockchain* requires transactions to be grouped together in blocks, each block to be cryptographically linked to one previous valid block (a block that includes transactions that have modified the distributed ledger), and each DLT node must process each block in sequential order. In contrast, a *directed acyclic graph (DAG)* does not group transactions into blocks. Instead, each transaction references other valid transactions (that have modified the distributed ledger), and each DLT node can process each transaction in different orders. Finally, a *block DAG* groups transactions into blocks, each block is cryptographically linked to other previous valid blocks, and each DLT node can process blocks in different orders.

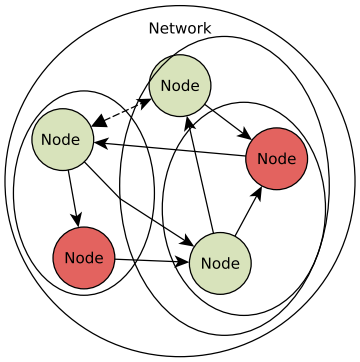
As regards collections of DLT networks, we say we have heterogeneous DLT networks when the technologies and their respective networks are different; we say we have homogeneous DLT networks when only the networks are different.



(a) DLT nodes run a DLT protocol. Different node implementations can communicate if they run the same DLT protocol and have the same DLT network configuration.



(b) DLT nodes can have different implementations of the same protocol (represented by different colours). A DLT network is a set of nodes connected in a mesh network manner.



(c) Networks can be organized into subnetworks. Nodes of each subnetwork share a partial view of the distributed ledger. Nodes can be in one or more subnetworks.

Fig. 1. Comparison between DLT nodes, DLT networks, and DLT subnetworks

Interoperability among DLT networks. Different DLT networks can connect to other DLT networks. An interoperability mechanism (IM), often called a bridge, can connect networks to other networks, subnetworks, or centralized systems. Figure 2 depicts the mental model on DLT networks, subnetworks, and interoperability mechanisms. DLT protocols instantiate DLT networks that, in its turn, can be connected to DLT subnetworks. subnetworks are more concerned with a specific scope (specialization of the network), i.e., they can focus on scalability, achieved, for example, via a different state model; or features. This figure considers the Carbon Emission Network (implemented with Hyperledger Fabric v2 and the Ethereum main net). The Hyperledger Fabric network contains two channels (subnetworks) that can

communicate with each other, but not natively. On the other hand, we consider the Substrate framework as the DLT protocol (or, more concretely, the SDK to generate blockchains), instantiated as the Polkadot network. Polkadot can connect to its subnetworks via the relay chain. A relay is a smart contract in a target blockchain that functions as a light client of a source blockchain. Light clients are network nodes that are solely part of the blockchain history, i.e., relevant transactions (vs. full clients that store the whole blockchain history). They can verify that a transaction was included in the blockchain, typically using block headers (e.g., via Merkle proofs [44]), but not validate it. Relay smart contract receives block headers from relayers, nodes that fetch blocks from the source blockchain and give them to the target blockchain. Relays behave like oracles (except that they have to process it instead of receiving the processed information). Blocks given to the relay smart contract can be contested by other relayers by presenting a Merkle tree proof. The relay chain acts as a relay, realizing the bridge between parachains. Each parachain can connect to the relay chain via a module called Cumulus, and send messages to other parachains by using a message format XCMP [107].

Vertical interoperability (from networks to subnetworks and vice-versa) and horizontal interoperability (between subnetworks and between networks of different systems) compose the spectrum of interoperability covered by this paper.

Horizontal interoperability can be implemented in a multitude of ways. Layer-2 solutions are independent DLT networks connected to other networks via interoperability mechanisms. They aim at solving scalability problems with the DLT networks they interoperate with. Scalability is enhanced by allowing the network to offload transaction processing and enabling new features from the original DLT network. Those subnetworks are pegged to the networks via cryptographic mechanisms [12, 115].

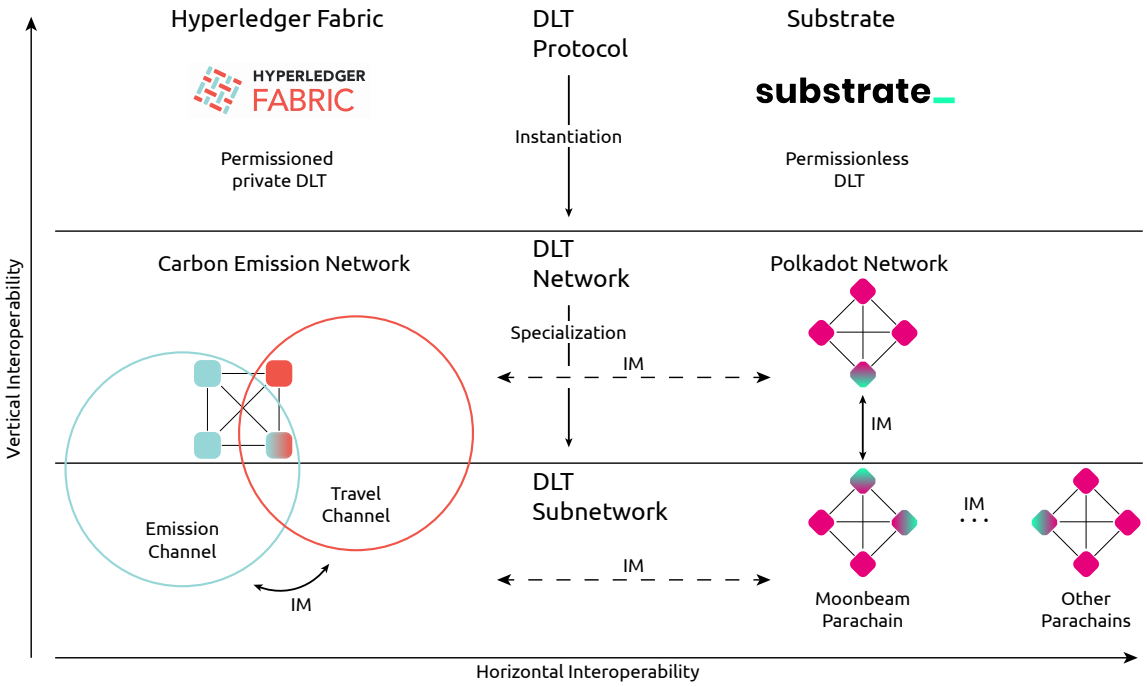


Fig. 2. DLT protocols, networks, and subnetworks.

Blockchain interoperability is challenging because it implies going beyond two different trust boundaries and establishing a new boundary. Network boundaries also influence state ownership: in a centralized system, the state is owned by a single party, and hence any party interoperating with such a system needs to trust it. A decentralized system, in its turn, defers the state ownership to the collective, where a protocol is used to update that state (and achieve *consensus*).

A new (trust) boundary is formed when two systems are interoperating. The trust assumptions of the new boundary can be lowered by systems to provide proof of state for a blockchain view[3] [3, 9]. The new boundary needs to assume that each ledger is secure. Security depends on the threat model and network assumptions. Generally, security properties such as safety assume an honest majority of participants and arbitrary Byzantine behavior. Furthermore, given those assumptions, each ledger can provide liveness and persistence [49]. Liveness says that all transactions originated by honest parties will eventually be included in the blockchain; persistence states that once a transaction is included in the blockchain of an honest party, it will be included in all honest parties.

Key Takeway 4. *Interoperability requires a trusted third party*

Cross-chain communication requires a trusted third party [137]. However, a trusted third party can be centralized or decentralized, being an example of the latter a blockchain (whereby its consensus is used as an abstraction for a trusted third party) [12, 137]. A prerequisite for the system to be a trusted third party is its safety, i.e., common prefix and chain quality [49]. A decentralized interoperability solution implies the usage of a blockchain consensus as a trust anchor for the solution.

Different trust assumptions exist for each ledger, e.g., at least one honest node in Hyperledger Fabric, or the majority of the computational power, in the case of proof of work blockchains such as Bitcoin. Thus, when choosing an interoperability solution, the users who participate in the origin DLT must trust the involved DLTs and the IM. Ideally, both should be decentralized [115].

DLT as a System Component. Several studies have presented blockchain as an infrastructure for data storage, and computation [89, 136]. Blockchain can be viewed as a system component that eases trust assumptions between mutually untrusting participants. DLTs can be accessed by centralized systems – and thus need software components that provide the necessary infrastructure for connecting with it (key management, secure connection, state storage). Interoperability across DLTs implies the existence of another middleware layer (another system component) that can bridge nodes.

2.2 Multiple DLT Decentralized Applications

DLT use cases are already in production, creating value [12]. As enterprises integrate blockchain in their business processes, the requirements will bring the need to use several types of DLTs. Decentralized applications (dApps) will then need to utilize multiple DLTs as their infrastructure, because one DLT cannot cover all use cases (i.e., offer the same functionality although there are different tradeoffs in security, scalability, and decentralization). We highlight two use cases that demonstrate the importance of this field, implemented by multiple DLT decentralized applications (mDApps).

Carbon Emissions. The first example is Hyperledger's Cactus implementation of the Carbon Emission App from the Hyperledger Carbon Accounting and Neutrality Working Group [28]. A detailed explanation of this use case can be

[3] stemming from the business process view integration research area, studying the creation, merging, and processing of views [9].

found in Hyperledger [27]. The purpose of this use case is to reward carbon emission reduction by orchestrating heterogeneous blockchains: one focused on data collecting, and another on the reward incentives.

A Hyperledger Fabric network collects emission records (activity data), e.g., energy consumption, travel mileage, and widgets produced. The emissions records are not continuous because both the emissions factors and the data for calculating emissions are based on long time windows (e.g., utility bills are produced each month). Periodically, the activity is aggregated to be later converted to an emission token (ERC-721). Emission tokens are created on Ethereum's public network from the collected data on Fabric to be traded against allowances that reward emission reduction. Figure 3 depicts this network.

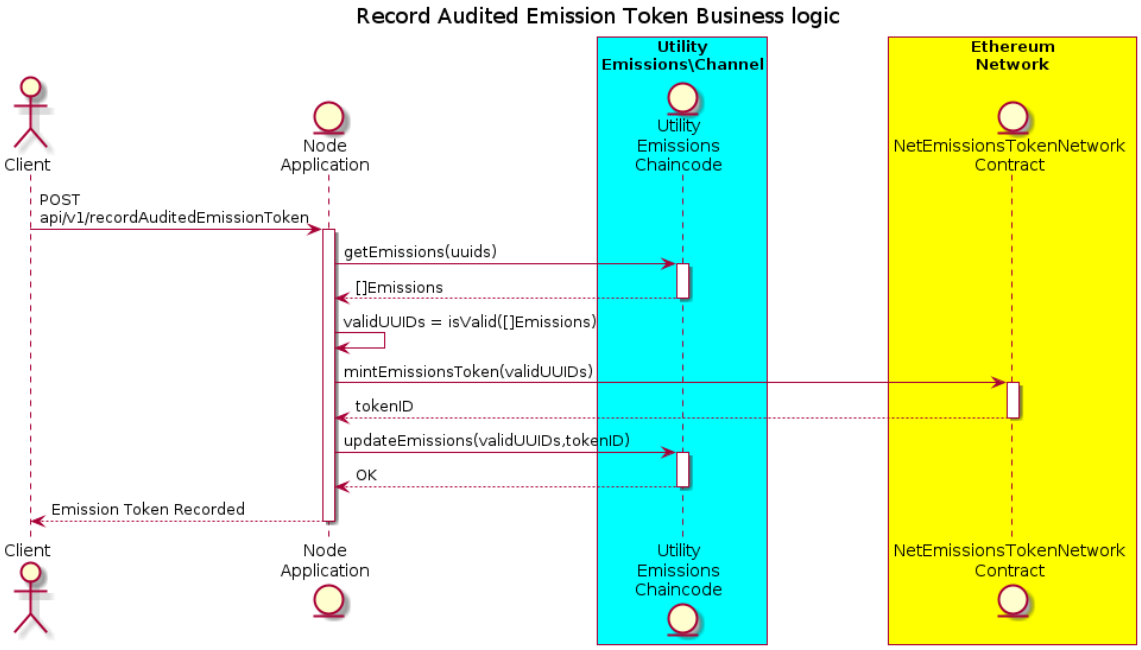


Fig. 3. Sequence diagram of the Carbon Emission use case.

This example contemplates a private, permissioned ledger used for performance and privacy reasons, but where the final output (carbon emission tokens) are stored in public blockchains as a reward. In particular, the performance of Hyperledger Fabric in terms of throughput and end-to-end latency is superior to most public blockchains due to its consensus and low number of peers. Privacy can be assured because only the peers involved can read the global state or if needed, only a subset of peers could read part of the global state (i.e., by utilizing channels or private data).

LACChain. LACChain [80] is a Global Alliance for the Development of the Blockchain Ecosystem in Latin America and in the Caribbean, led by the Innovation Laboratory of the Inter-American Development Bank Group (IDB LAB) in cooperation with partners and strategic allies.

LACChain aims to provide infrastructure and technical tools, on top of the three layers the LACChain network comprises (DLT, self-sovereign identity, and tokenized money) that are useful for developing applications with social impact that contribute to the development of the countries of the region.

LACChain aims to provide a community and a general-purpose infrastructure for the realization of several use cases, such as supply chain, cross-border payments, and financial inclusion. This project currently utilizes two blockchain technologies, Ethereum and EOS [114].

Quant [113] and LACChain are piloting a project to provide private currency payments between retail customers from different financial institutions. This prototype involves retail customers creating transactions on a public permissioned Ethereum network. These payment amounts between the customers are tallied, even though the identities involved in the transactions are hidden via zero-knowledge proof technology. As all payments are recorded, currency exchanges between financial institutions can be netted and settled efficiently on a separate private permissioned Ethereum network. The blockchain interoperability solution infrastructure to allow this netting and settlement to occur is Quant's Overledger, where the related interoperability applications are built on top of it utilizing Overledger's cross-DLT standardized data model.

3 STATUS QUO OF BLOCKCHAIN INTEROPERABILITY

Throughout this paper, we summarized and built on top of existing knowledge of blockchain interoperability, including existing solutions, challenges, and opportunities.

3.1 Interoperability Layers

This section studies the main interoperability layers based on existing interoperability platforms: we pave the way for systematically analyzing IMs.

Interoperability among computer systems is typically defined in terms of several layers [67]. Although it is possible to come with a detailed architecture for interoperability applications, interoperability has different meanings (and thus uses different techniques) depending on its domain (e.g., for European Union states [45], for language resources [67], supply chain [29], governments [57], and others). Hence, we adopt the European Interoperability Framework model [45] from the European Commission. This model is based on four layers, as depicted in Figure 4.

- Technical interoperability: links systems and services by adopting compatible data formats, communication protocols, interface specifications, integration services [45]. Information exchange is achieved with technical interoperability, but no guarantees on interpreting the received information, but there are no guarantees on how the received information is interpreted.
- Semantic interoperability: exists when systems can interpret information following a defined ontology (i.e., following a well-known model for information). As a consequence, information from one system can be interpreted in another. Some prerequisites of this type of interoperability are agreements (or conventions) on data formats. Protocol messages (and the protocols themselves) and the representation of assets are part of this layer. Thus, semantic interoperability subsumes information syntax (what is the information format?) and information semantics (what does the information mean?).
- Organizational interoperability: concerns aligning the requirements and interests of the user community by leveraging cooperation and integration of business processes between organizations via arrangements and protocols, typically under a formal or semi-formal deal.
- Legal interoperability: ensures organizations can cooperate under "different legal frameworks, policies and strategies" [45]. This includes a certain degree of coherence between legislations so that the assets managed under the semantic interoperability layer can be managed consistently.

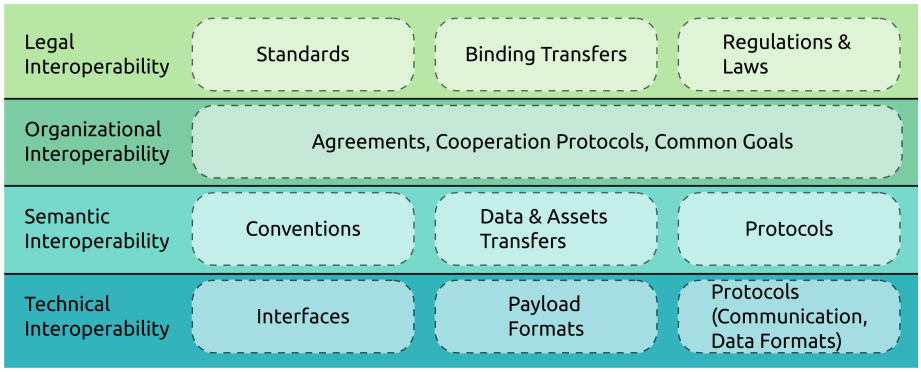


Fig. 4. Interoperability layers (see [45])

The orchestration of the four layers (arguably, at least the first two) could lead to a seamless integration of DLTs, leading to value exchange. For instance, if there is no regulatory framework that ensures the validity and legality of a cross-jurisdiction asset transfer (at the legal interoperability layer), organizations may not cooperate seamlessly (organizational interoperability layer). These incompatibilities affect different viewpoints, according to stakeholders' *concerns* [36]. Four concerns are proposed on the Framework for Enterprise Interoperability (FEI) [71]. The *business concern* regards barriers in organizations to cooperate despite differences in the decision-making process. The *process concern* regards how various artifacts that support the business (processes) work together. The *services concern* identify the applications and their interfaces that support processes. Finally, the *data concern* regard data management from different supports. Each concern is related to all interoperability layers, and each layer is related to one another (typically following a bottom-top approach).

It is worth noting that other frameworks are equally valid, such as the Cloud Interoperability Standard ISO/IEC 19941:2017, being currently studied by ISO's WG7 (interoperability). In this framework, three layers exist: technical, business, and governance. For the sake of granularity, we choose the European Interoperability Framework model.

As an example, let us consider a cross-jurisdiction DLT-backed asset transfer [10]. Technical interoperability allows exchanging bytes across systems; semantic interoperability allows exchanging the asset – running a protocol creating entries representing ownership on both ledgers. Organizational interoperability concerns the deal between institutions that want to arrange digital asset transfers. Legal interoperability assures the validity/legal character of the asset. The last layer requires coordination between legal frameworks and, possibly, between the interoperability solution and the current regulatory framework.

Key Takeway 5. *Legal and organizational efforts are lagging behind*

The lack of legal interoperability, in the form of standards and IMs hinders the development of dApps. Although governments and enterprises are interested in the technology alike, there is a gap between its potential adoption [12].

3.2 A Model for Interoperability Solutions

In this section, we present a generic interoperability framework, as the blueprint to design an IM.

When connecting an application to one or more DLT networks, a developer (or software architect) has a few choices. Firstly the application could connect directly to a DLT node. Secondly, the application could connect to multiple DLT nodes, but then the complexity of managing the routing between the DLT nodes needs to be contained within the application. Instead and thirdly, the application could connect to a DLT node proxy which handles the routing and load balancing issues creating logical separation. The final option for an application is to connect to a DLT network via a DLT gateway, or a combination of gateways. DLT Gateways can implement data and asset transfers, as well as asset exchanges and come in many forms.

An IM is an application (classified as an oracle or cross-authentication, can be executed off-chain, on-chain, or both) that includes a connection mode (e.g., a DLT Gateway) and performs an interoperation mode. IMs provide access to its functionality via an API (we defer a formal definition of an IM for future work). The functionality an IM provides is to execute cross-chain logic via a set of protocols. The processing occuring in an IM can be persisted in a storage, composed of a local state and a cross-chain state. The local state stores all relevant data for local computation (e.g., processed output from business logic plugins, logs) and cross-chain state (joint state representing relevant computations performed over multiple systems).

Figure 5 illustrates a model for interoperability solutions. The IM exposes its functionality via a set of APIs. The APIs redirect the requests to the responsible module handling specific functionality upon invoked. Those modules are called cross-chain rules, cross-chain logic, or business logic plugins (BLPs). Cross-chain logic modules process the request, translating it into transactions or requests to external systems, including DLTs. This processing can be persisted in storage. Protocols support the execution of cross-chain logic by acting as a middleware layer between high-level logic and specific interactions with other machines or DLT transactions. Eventually, the interaction with target nodes (either DLT nodes or other IMs) receives a response back, which is processed and optionally persisted. The processed responses can be redirected to an external system. A cross-chain state can be built from executing cross-chain logic to implementing cross-chain protocols. That state can be shared with multiple instances of the same IM, or another one.

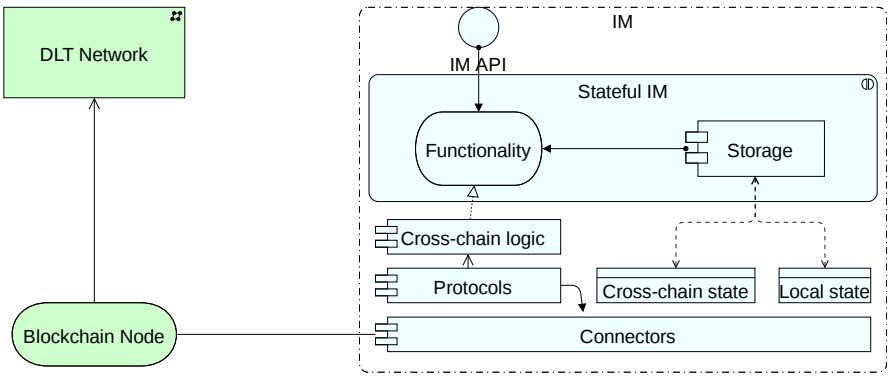


Fig. 5. Representation of a general purpose blockchain interoperability solution in the Archimate modelling language [122].

This conceptual architecture effectively implements the technical and semantic layers of an IM. The organizational layer comprises how organizations cooperate across trust boundaries to achieve common goals in agreements valid on that trust boundary. In the newly formed trusted boundary (trust boundaries 1 and 2), the IM can interoperate with other systems (e.g., centralized systems) following specific protocols that realize cross-boundary cooperations. The

legal layer applies to all trust boundaries. In particular, the applicable law varies according to the specific jurisdiction and norms. Figure 6 represents an IM connecting three different trust boundaries.

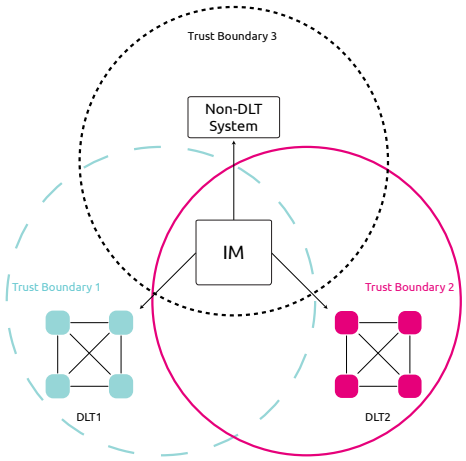


Fig. 6. Trust boundaries between a non-DLT system, two different DLT networks, and an IM.

We call an IM *trustless* if two conditions hold:

- *verifiable correctness*: there is a method to check that the IM runs a well-defined *functionality* (e.g., protocol, arbitrary business logic) among chains. Misbehavior in the executing environment can be detected and thus held accountable. This implies that actions performed by the IM need to be stored, preferably in a public forum.
- *eventual data consistency*: all verified instances of the IM will eventually return the same result for any specific function call with the same input. However, a single IM may return to the user the result of a cross-chain transaction only when it is finalized, i.e., all sub-transactions have been committed. This stronger consistency guarantee can be provided at the expense of latency.

3.3 Blockchain Interoperability Solutions

Choosing a blockchain interoperability solution requires asking at least two questions: "what do you want to connect", and "how does the interoperability solution connect the systems?".

3.3.1 Interoperation Mode. The "what" question concerns the artifact managed by the blockchain interoperability solution, i.e., the interoperation mode. The artifact exchanged can be data or assets. Data are arbitrary byte strings representing a piece of information on the blockchain (technical layer). It could be a key-value pair, metadata about the blockchain. Data can be copied from blockchain to blockchain.

Assets can be represented in the technical layer (by a string, for example). However, in the semantic layer, they "take form" by representing a fungible or non-fungible value which is or is not linked to a physical identity (in case it is, it is called a *digital twin* [110]). Therefore, they should not be copied among DLT networks but rather be *transferred under specific conditions*. More specifically, an asset transfer should abide by the rules of each DLT (e.g., no double spend), i.e., preserve at all moments the invariants of all the DLTs it affects. In DLTs, double spend can occur when an attacker sends tokens for a pending payment in a transaction to a victim in return for a product. The victim releases the product.

Then, the attacker cancels the pending payment transaction, such that the original token transfer does not become recorded on the blockchain ledger (and thus their tokens are preserved). In the context of interoperability, double spend can happen in cross-chain asset transfers, when a lock/burn on the source DLT (the DLT in which the transaction is initiated to be executed on a recipient DLT [12, 59]) was not performed. The same representation of an asset could be valid in several DLTs, leading to a *new type of double spend*. Solving this problem implies synchronizing DLTs at the semantic layer, i.e., the involved blockchain interoperability solutions need to run the same protocol that prevents invariants from being violated, with on-chain notarization (e.g., via smart contracts).

The interoperation modes are:

- *Data Transfer*: data is copied from one DLT to another, with an optional intermediate processing step. For example, copying price information from one DLT into another [3].
- *Asset Transfer*: unilateral or bilateral asset transfers. Assets are transferred from one DLT network to another (implies burning or locking the asset on the source DLT network). Tokens that are minted are typically called wrapped tokens [53], because its value is anchored on another asset. For example, locking Bitcoin to a multi-signature address on the bitcoin blockchain and minting a representative asset on the Ethereum blockchain (such as wBTC [98]).
- Asset Exchange: atomic asset transfers. Assets are exchanged in their respective DLT network, i.e., no transfers across DLT networks occur. Participants need to be present in both chains for this exchange to happen. For example, swapping promissory notes across two different distributed ledger systems between two users where both assets just change the address on their native DLT network [10].

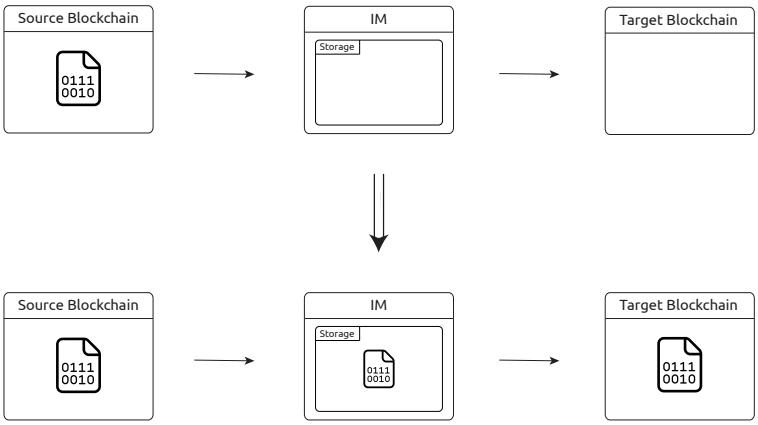
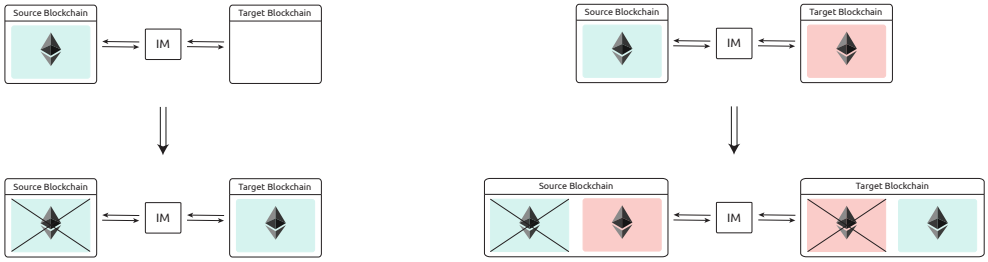


Fig. 7. Data transfer between two DLTs.

Figure 7 represents a data transfer from the source blockchain to the target blockchain. A blockchain interoperability solution requests data from the source blockchain and writes it on the target blockchain. Data can be copied. As blockchains increasingly comprise more value, represented by assets, value transfer among blockchains needs to be handled carefully, as it exposes a new class of attacks: cross-chain attacks. In cross-chain attacks, attackers attempt to double spend an asset by manipulating cross-chain protocols. Assets are then more sensitive to manage in terms of interoperability.



(a) Unilateral asset transfer between two DLTs.

(b) Bilateral asset transfer between two DLTs.

Fig. 8. Asset transfers

Figure 8a represents an asset transfer from the source blockchain to the target blockchain. A blockchain interoperability solution requests data from the source blockchain and writes it on the target blockchain as an asset (semantic layer protocols are required). Thus, the IM needs to make sure the representation of the asset on the target blockchain is changed to used (or burned). This implies the IM needs to check for conformance on both DLTs. Figure 8b represents an asset exchange (bidirectional asset transfer, or two asset transfers) between both blockchains. Again, the IM needs to check that both blockchains are in a consistent state. Note that these figures are high-level and hide details. More detailed procedures show the rules for asset transfers in the next section.

Cross-chain asset exchanges can also be classified into two types, permanent and temporary:

- *Permanent asset exchange*: assets are exchanged between parties with no obligation to reverse the exchange later. e.g., hash time lock contract swaps[4].
- *Temporary asset exchange*: assets are exchanged between parties where the conditions to reverse the swap are in place. Examples include a cross-ledger loan where a user places an asset into a smart contract on one chain for another asset to be borrowed on another chain [19].

In asset transfers, it is the responsibility of the interoperability solution to establish a new boundary of trust for both previously established boundaries. Since one asset transfer will typically involve several transactions (from the source and target blockchains), it is desirable to do so via atomic cross-chain transactions. Atomicity is desirable because, in the case not all transactions are completed, the union of systems might be left in an inconsistent state (although several solutions exist, such as rollback [11]).

3.3.2 Connection Modes. There are three methods for a dApp or mdApp to connect to a DLT. These mechanisms are called the *connection modes*. Those are:

- **DLT Nodes**: DLT nodes are the software systems that run a DLT protocol. The application could connect directly to a DLT node. While anyone can run their single DLT node, this is not crash resilience and not scalable from a load balancing perspective[5]. Example: an Ethereum node being run locally (Geth client).
- **DLT Proxy**: a DLT node proxy manages the routing and load balancing issues between an application and one or more DLT nodes, creating logical separation. To an application, interacting with a DLT node proxy is nearly identical to interacting with a DLT node as the message requests and responses will be virtually the same. The

[4]A hash lock is an artifact that requires a preimage of a hash to trigger behavior. More concretely, a hash lock protocol relies on the preimage resistance property of a hash function H, such that hash = H(secret). A timelock is an artefact that triggers the ending of a protocol when a certain time has passed (e.g., in terms of the number of blocks). A hash lock time contract combines these concepts to realize a timed, programmable escrow supported by a DLT
[5]Some enterprises provide a DLT Proxy service (node as a service) to solve this problem.

only possible difference may be identifying metadata in the messages to track the DLT node proxy users (e.g., for rate limiting reasons). Examples: a group of permissionless network nodes runs on Kubernetes (self-hosted or run by a third party). Some enterprises provide a DLT proxy, such as Infura's Ethereum DLT node as a service [68], or Blockdaemon [15].

- **DLT Gateways**: Like a DLT node proxy, a DLT gateway also manages the routing and load balancing issues between an application and one or more DLT nodes, creating logical separation. Example: Polkadot's block explorer; Self-hosted ODAP gateways; Quant Network's Overledger.

DLT gateways are not DLT nodes. Instead, they are a collection of services built around DLT nodes, as Figure 9 shows. The services it provides (i.e., the protocols it runs) and identity management, access control, security, and liveness properties are left to be instantiated by the DLT gateway administrator. An example of a gateway is the ODAP Gateway [11, 61]. DLT proxies can also promote geographical diversification, alleviating some known blockchain cyberattacks (e.g., eclipse attack). However, it could be desirable to run non-native business logic from DLT nodes.

A DLT gateway provides additional non-DLT functionalities. Such additional functionality can include data analytics or complicated cross DLT network processes.

It could be desirable to use a DLT gateway instead of a DLT node or DLT node proxy if the gateway's additional services are crucial to your application or to smooth your application build process. For instance, a DLT gateway could utilize a standardized data model, meaning that, unlike DLT node proxies, a DLT gateway can be used to connect to many DLT networks of multiple DLT types. On the other hand, a DLT node proxy could be more desirable if the application developer is experienced and very familiar with the underlying APIs and data models of the DLT nodes.

Both DLT gateways and DLT node proxies can promote geographical node diversification. Additionally, DLT gateways and DLT node proxies can promote technical node diversification (e.g., running multiple different DLT node implementations for a particular DLT, such as geth and nethermind Ethereum nodes). This minimizes the risk of application failure if there is a bug in a particular node implementation.

However, it could be desirable to connect an application directly to a DLT node, for permissioned DLT networks where the attack vector of DLT nodes is significantly less and therefore running a DLT node in a container orchestration system (e.g. Kubernetes) would suffice.

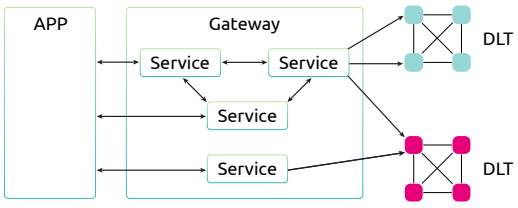


Fig. 9. Gateway architecture. A gateway provides a set of services to a client application, while connecting to different DLT networks.

3.3.3 Solution Categories. Categorization of DLT interoperability solutions attempt at answering *How is interoperation achieved?*, and what are the trust boundaries each solution creates. In the previous section, we presented the connection modes. In this section, we present a unified IM categorization that considers the connection mode, interoperation mode, and trust assumptions required by the solution category. Contrary to common knowledge, there are only two non-intersecting IM categories. All interoperability mechanisms are created from the following categories of solutions.

In particular, transferring data requires oracles, and transferring or exchanging assets can be done in two ways: 1) locking an asset in a source DLT, and creating its representation on a target blockchain, implying transactions on both DLTs – using oracles – or 2) via native transfer transactions (Alice transfers to Bob asset A in DLT X for asset B in DLT Y) – cross-authentication.

Solution Type 1 (Oracle).

Oracle interoperability solutions allow a DLT system to make use of external data from another system [25, 42, 95], increasing the connectivity of DLT-based applications. There is a lot of on-going research on the security and fairness of interoperability via oracles. In particular, oracles could be selecting certain transactions to be included in the target blockchain for its own benefit, similarly to the miner extractable value problem[38]. We do not aim to cover such topics in this work, nor to present oracles in great detail, as this is well covered in the literature.

Code deployed into a distributed ledger cannot access external resources or data without the help of an intermediary. These intermediaries (known as oracles) gather external data, placing it into transactions that are subsequently added to the distributed ledger, therefore allowing this retrieved data to be read by deployed smart contracts. Note that oracles can fetch data from a non-DLT system or another DLT-system. Oracles are classified into two types [95], pull-based, or push-based.

The parties involved are the user and its smart contract (deployed on a DLT), the oracle and its smart contract, and external systems (decentralized, centralized).

- Pull-based oracle data transfers: upon request, those fetch data from off-chain systems and send the data to a DLT (via a transaction). These oracles operate differently depending on the transaction execution model of the DLT:

(1) order-validate-execute model (e.g., Ethereum): pull-based oracles on these DLT systems require multiple transactions to complete the data request process. An example would be the following: a smart contract issuing two transactions: 1) a transaction to the client smart contract which triggers a call to the oracle smart contract. This call details the data it wishes to obtain from an external system; 2) the oracle observes this request and creates a transaction with the necessary data, which is sent to the oracle smart contract called by the client. This way, the client smart contract now has the necessary information accessible via the oracle smart contract. This model could require one transaction if an oracle smart contract already holds the necessary data (requiring the oracle client to be pushing data periodically, i.e., the oracle is a push-based oracle).
Figure 10 represents a pull-based oracle operating with a order-validate-execute DLT. The client smart contract calls the oracle smart contract with a request for information (step 1) the latter does not have (for example, the request from the client smart contract includes a GET HTTP request). The transaction is recorded, and the oracle listens to transactions that call its smart contract, via an off-chain client (step 2). Upon recognizing them, it performs the requests by collecting information (steps 3-6). Upon eventual processing (step 7), the information is pushed to the oracle smart contract (step 8). The oracle smart contract now has the necessary information in its storage (or memory) (step 9). Finally, the oracle smart contract returns the information to the the client smart contract (step 10).

(2) execute-order-validate model (e.g., Hyperledger Fabric): pull-based oracles on these DLT systems only require one transaction to complete the data request process as nodes involved in the execution and verification process perform the oracle functionality by fetching the data themselves. Before the transaction is confirmed,

meaning that the external data can be immediately used by the requesting transaction (e.g. in the case of Hyperledger Fabric chaincode HTTP requests or Corda flows that include an Oracle node).

- Push-based oracle data transfers: obtain external data without an explicit request from a DLT transaction. These oracles usually add the data into smart contracts to be easily consumed by other smart contracts (via smart contract to smart contract calls).

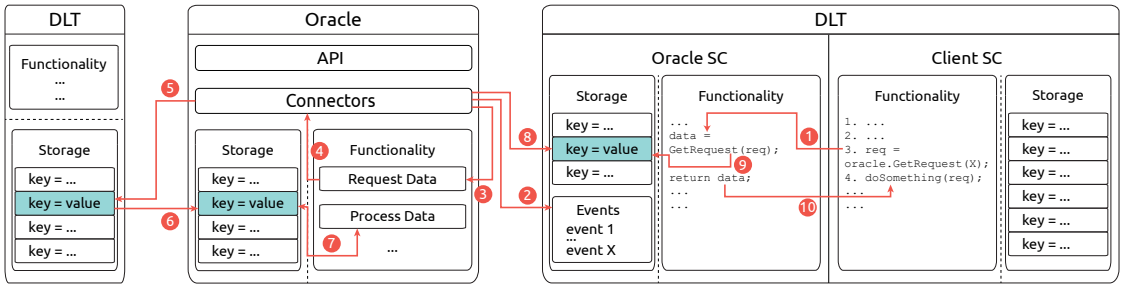


Fig. 10. Pull-based oracle architecture on execute-order-validate (e.g., Ethereum)

Oracles inject information into a trusted network. This information can be used for decision-making by nodes, meaning oracles greatly influence the subsequent state of the network as a whole. It means that oracles are trusted third parties. The trust model of oracles might vary, from voting-based oracles to reputation-based oracles to a single trusted oracle [101]. The choice of an oracle then boils down to the choice of functionality offered, weighted with the acceptable level of decentralization of that oracle network.

Note that single DLT network oracles also exist (sometimes referred to as on-chain oracles, meaning located, performed, or run inside a blockchain system [72]). These oracles (when triggered by a transaction) collect data from different on-chain sources and present it in an easy to consume format [46]. But these oracles do not aim at providing cross DLT network interoperability [46].

Oracles also support asset transfers across chains (sometimes known as bridges), typically unidirectionally, with mediated off-chain communication. These transfers are considered by commit-and-execute protocols [90, 138]. In this scheme, an asset residing in the source DLT network is burned (i.e., deleted) or locked, and a representation of that asset is minted (i.e., created) in the recipient DLT network. Both stages are implemented with smart contracts that implement specific rules for these operations.

Asset transfers via Oracles can occur using both push and pull based oracles.

- Pull based oracle asset transfers: these are started by a transaction to a certain address or smart contract on a source DLT. This transaction contains the instruction to lock or burn the asset on the source DLT. Now an off-chain party, for example, the beneficiary of the transaction on the destination DLT (or an off-chain trusted third party) watches for the confirmation of the transaction on the source DLT. It sends a proof of that transaction to the destination DLT (for example, a Merkle proof). More specifically, the proof is sent to the smart contract on the target destination DLT, along with information such as the beneficiary's address. This mechanism triggers the smart contract to mint and sends tokens to that address. At the end of the procedure, the number of assets is preserved.

- Push based oracle asset transfers: these are started by a transaction on the destination DLT. This transaction contains the instruction to lock or burn the asset on the source DLT, create (or redeem) a representative asset on the destination DLT and this transaction provides the related proof of ownership of that asset on the source DLT (such as a signature). In some interoperability solutions, this single transaction may be enough to complete the asset transfer (e.g. when a payment channel or Zero Knowledge Proof Rollup is the source DLT). Whereas in other interoperability solutions, other transactions, on the source DLT and possibly again on the destination DLT may be required to complete the asset transfer.

Optionally, cross-ledger audit trail information can be included in these transactions. Note that it is technically possible for token owners to operate their bridge, but this generates a problem around the general acceptance (e.g., by exchanges) of the token minted on the recipient DLT network. Some protocols that perform asset transfers include services to monitor the process for their users.

Interoperability Mode:

- Data Transfer (typically called oracles)
- Asset Transfer (typically called bridges)

Trust Assumptions:

- Centralized (single oracle)
- Decentralized (oracle consortium)

Advantages:

- Anyone can run their own oracle provided access to the recipient DLT.
- Oracles can allow the information collection phase to be separated from the processing phase, allowing arbitrary processing.
- Push-based oracles can send already processed data (i.e., ready to consume) or raw data. Processed data allows the amount of on-chain processing to be minimized.
- Pull-based oracles can allow a more transparent audit trail regarding who requested the data and who collected the data, as these actions are recorded inside distributed ledger transactions.
- Some DLTs allow smart contracts that can call external systems (e.g., Hyperledger's Fabric chaincode), i.e., the nodes running the protocol have the ability to perform as an oracle. This mechanism allows decentralized oracles, as oracle calls are made on-chain and are under consensus scrutiny.
- Oracles can be designed to declare who has permission to operate as an oracle. For instance, an IM can be designed to allow anytone to operate as an oracle (permissionless oracle system). In this case, there should be a mechanism that allows suspected invalid data to be challenged, and its authors possibly penalised. Alternatively, only certain oracles could be allowed, in which case a permissioned oracle system is used.

Disadvantages:

- Oracles support asset transfers, but trust needs to be put in the oracle group and the semantic layer supporting such transfer.
- Pull-based oracles can require multiple transactions for more generic calls (e.g., for generic HTTP requests vs. smart contract calls), raising the latency of the solution.

- Availability of oracles is a deciding factor, as smart contracts may rely on them to provide accurate information in real-time. Failures in oracles (either crash faults or Byzantine faults) can occur and originate great losses (e.g., attacks on oracles DeFi [46]).
- Data is fed via DLT transactions that implies a minimum delay in the order of a few seconds to a few hours (if one considers finality).
- A smart contract depending on an oracle implies the trust of a (probably) smaller set of parties, compared to the number of nodes of the DLT network enforcing the correct execution of the smart contract. This weaker trust assumption is an attractive target for attackers.
- Can require synchronizing with off-chain parties (e.g., other exchange parties or the bridge operators).

Examples:

- Chainlink [18] (pull-based and push-based): provide external information to a set of smart contracts that can expose that information to other smart contracts for a fee.
  ChainLink selects qualified data feeders to provide data expected to represent the ground truth. Data feeders aggregate data via decentralized selection through staking their reputation (represented by LINK tokens). Data aggregation is done via statistical measures.
- BTC Relay [44] (push-based): provides information from the Bitcoin network to the Ethereum network.
  BTC Relay is a smart contract on Ethereum that stores block headers from the Bitcoin network. Nodes called relayers obtain the headers and send them to the BTC Relay smart contract. Smart contracts on the Ethereum network can then utilize information from the Bitcoin network by providing a Merkle proof referring to a certain block header.
- Polkadot (interoperability modules): the Polkadot ecosystem has several bridge projects allowing to connect ecosystems [106]. These mechanisms mostly allow unidirectional asset transfers (albeit two unidirectional transfers can be done, realizing a bidirectional transfer), effectively connecting assets from different chains. For example, Snowfork is a general-purpose bridge between Ethereum and Polkadot. This will enable not only ETH to be transferred from Ethereum to Polkadot, but also ERC20 assets.

Solution Type 2 (Cross-Authentication).

Cross-authentication interoperability solutions allow parties to exchange assets across DLTs, where each party sends a transaction on each DLT. To this end, parties need to authenticate on both chains to perform the transfers. This process typically happens without a trusted third party, as what is needed is some off-chain synchronization to set up the transactions to happen in both chains.

Trustless Asset Exchanges correspond to one asset exchange with non-mediated communication. The de-facto method for implementing this scheme are Hash Lock Time Contracts (HTLCs), where both parties deploy a smart contract in each chain that transfers the right amount of coins to the other party.

HTLCs consist in facilitating an asset exchange between two parties (typical case, although multi-party HLTCs exist [12, 64]) on a different blockchain (both parties can access it). Party from DLT 1 (P1) creates a secret s such that the hash of the secret, h(s) is put in a smart contract on DLT 1, transferring an asset to P2. The contract is hash locked with s, and a timelock t. Thus, P2 can redeem the assets from DLT 1 with secret s until time t. Upon confirming that the contract is correctly instantiated, P2 can create a smart contract on DLT 2 with the same hashlock h(s) but a timelock t' < t. The smart contract sends assets to P2 from DLT 2. This ensures that P1 can redeem assets before P2, with a slack

t-t'. When P1 asserts that the contract from P2 is published, that party can send secret s, redeeming its assets. P2 now holds secret s, and can use it to redeem its assets on DLT 1.

HTLCs imply handling the technical layer (the hashing functions that are used to construct the secret needs to be supported by the involved DLTs) and the semantic layer (the information exchanged has meaning – assets – and needs to preserve a set of rules on both chains – avoiding double spending, for instance).

On the other hand, centralized asset exchanges are cross-authentication solutions that allow asset exchanges. These exchanges (also called notary schemes [12]) are a legal escrow that exchanges assets from one DLT for assets from another DLT. Decentralized exchanges are typically facilitators of such transactions by offering a bookkeeping system that matches buyers and sellers. Although decentralized exchanges running in heterogeneous DLTs are appearing [96], the mechanisms used for interoperation are classified as oracles. An overview of how centralized and decentralized exchanges work is present here [12].

New types of HTLCs are showing up. In Hyperledger Cactus [93], the *cactus-plugin-htlc-eth-besu-erc20* allows to automatically deploy HTLCs on Ethereum via Hyperledger Besu. A similar package could automatically deploy two contracts: one in a permissioned DLT, and another in a permissionless DLT. As long as the parties exchanging assets are present in both networks, this scheme would work.

Interoperability Mode:

- Asset Exchange

Trust Assumptions:

- Centralized (centralized exchanges, notary schemes)
- Decentralized (HTLCs)

Advantages:

- Decentralized exchanges allow trustless asset exchanges between parties, by anchoring the correct operation of the process on the blockchain consensus).
- Platforms to create HTLCs running on heterogeneous DLTs, such as [93] streamline the process of setting up an exchange, diminishing the need for decentralized exchanges (and thus avoiding fees).

Disadvantages:

- Asset exchanges require multiple DLT transactions, which implies a minimum delay in the order of a few seconds to a few hours (if one considers finality).
- Network delays might render the execution transactions useless, wasting time and possibly transaction fees. However, some modern solutions eliminate this need.
- Trustless approaches require some off-chain coordination between users wanting to exchange assets. Decentralized exchanges simplify this process at the expense of some decentralization.

Examples:

- Hyperledger Cactus cactus-plugin-htlc-eth-besu HTLC: Cactus provides a package that can deploy hash time lock contracts on Ethereum via Hyperledger Besu. The package provides functionality to deploy initialization, refund, and monitoring endpoints. Aditional functionality (such as mediating off-chain agreements between the users of the HTLC) can be built on top of this package (i.e., a business logic plugin).

	Oracle	**Cross-Authentication**
Interoperability Mode	$\mathcal{D}$, $\mathcal{A}_t$	$\mathcal{A}_e$
Common Connection Mode	DLT Gateway	DLT Node, DLT Proxy
Can be used to build general-purpose use cases (vs. only transferring assets)	✓	✗
Native DLT security assumptions are enough to enable interoperation	✗	✓
Easily decentralizable	✗	✓
Easily implementable	✗	✓
Can parties be offline for interoperation to happen?	✓	✗(for HTLCs)

Table 2. Summary comparing oracles and cross-authentication BISs. $\mathcal{D}$ stands for data transfer, $\mathcal{A}_t$ for asset transfer, and $\mathcal{A}_e$ for asset exchange

- Exchanges: with a centralized exchange, users deposit fiat or cryptocurrencies in a platform that is used to swap for other assets. It does not require all parties to authorize transactions on both chains, but only requires the sending user and the exchange to authorize the swap transaction.

Summary. Contrary to common knowledge, there are only two non-intersecting categories. We emphasize that this is a general overview. In-depth descriptions of the protocols and its implementations can be found in [12] (both), [25, 42, 95] (oracles) and [137] (cross-authorization). Table 2 summarizes the studied categories.

Oracles can perform data and asset transfers, typically using DLT gateways. This is due to the ability of gateways to process data to the format the oracle smart contracts accept. Oracles can enable general-purpose interoperability, thus they have implementation overhead, as well as decentralization overhead. On the other hand, cross-authentication solutions are used for asset exchanges only - a DLT node or DLT proxy suffice. its implement but the latter only exchange assets.

The immense variability of IM solutions stems from the fact that many design patterns are built on top of those two categories. A detailed study on the available design patterns for IM is left for future work.

Key Takeway 6. *There is no technical distinction between "Layer 1" and "Layer 2" solutions*

The industry typically classifies DLTs into layer 1 infrastructure or layer 2. While layer 1s are standalone DLTs, layer 2s are DLTs extending the capabilities of layer 1s, attempting to solve, for instance, the scalability problem [12]. Both layer 1 and layer 2 solutions are groups of nodes (i.e., chains) running a protocol. Some of the node groups might anchor their security on another one (typically layer 2 chains share security [50] or re-utilize work [83] from layer 1s). This implies that layer 2s are better viewed as separate networks (technically similar to layer 1s) connected by an IM (typically called bridges).

4 WHICH BLOCKCHAIN INTEROPERABILITY SOLUTION DO YOU NEED?

Few studies provide guidelines to improve interoperability in blockchain solutions [8]. Specifically, there are no frameworks to evaluate cross-chain solutions in terms of interoperation capabilities systematically, performance, security, cost, and user friendliness. In this section, we put forward a first effort, based on our recent work [91]. Since standardized approaches to evaluate interoperability are needed [94], we propose our interoperability assessment

framework for DLTs. We start this section by presenting our interoperability framework, allowing the end-user to assess the current state, in terms of interoperability, of their DLT-based solution. After that, given that an initial assessment has been conducted, we present our framework to choose the infrastructure and functionality of an IM. Users can then improve the interoperability of their solutions by picking a suitable IM, based on the proposed decision models. Finally, one can re-evaluate the interoperability of their DLT-based solution by running the interoperability assessment again.

4.1 Interoperability Assessment

The goal of the interoperability assessment is to provide concrete, systematic guidelines for solutions to be compared in terms of interoperation capabilities – a concept called Interoperability Assessment (INAS) [36]. This work focuses on evaluating interoperation capabilities (and performance, to a lower extent). To this end, we propose three assessments. Each assessment defines a set of criteria, where each item yields a score. The score of all criteria outputs the score of that assessment. Summing the score of the three types of assessments yields the final score for the interoperability assessment. The higher the score, the better the interoperability capabilities of such a solution. Table 3 shows the score for each criteria. There are three assessments a system can take to measure the ease of interoperability regarding external systems [37]:

- *Potentiality assessment*: this assessment evaluates the maturity of a system to adapt to other systems. It answers the question *can the system interoperate with other systems as is*?
  This assessment provides an understanding of which infrastructures a DLT-based solution can connect to. The score for this assessment is divided into four categories, for a maximum of 4 points.
- *Compatibility assessment*: this assessment evaluates the interoperability between two known systems before or after changes to interoperation capabilities of both. It answers the question *how well can a pair of systems interoperate*? And *what are the current problems or barriers that prevent the systems from interoperating better?*
  This assessment provides an understanding of the "capabilities" the interoperability mechanisms offer (can it make two DLTs understand each other? can it comply with rules and laws?). The score for this assessment is divided into three categories, for a maximum of 3 points.
- *Performance assessment*: this assessment evaluates the interoperation processes during runtime concerning cross-chain transactions key metrics. It answers *what are the values for the interoperation metrics cross-chain latency, cross-chain throughput, and cross-chain costs?*
  The score for this assessment is divided into two categories, for a maximum of 3 points.

4.1.1 Potentiality assessment. The potentiality assessment evaluates technical interoperability (see Section 3). It takes a system based on a DLT protocol and evaluates its maturity towards interacting with other systems (requesting/providing data). Four levels of interoperability exist (c.f. Figure 11), in increasing order of complexity:

- *Level P1*: interoperation across different functionality (e.g., smart contracts) on the same subnetwork can happen. An example is smart contracts calling other smart contracts (on the Ethereum network, on the same Hyperledger Fabric channel). Even though level 1 interoperability may seem standard for some DLTs, e.g., smart contracts on Ethereum and Fabric, this is not the case for all DLTs. For instance, on Corda, Cordapps are deployed onto certain nodes. Each Cordapp includes a set of smart contracts used for transactions relating to this Cordapp. Allowing smart contracts created in one Cordapp to be utilized by transactions created in another Cordapp is a non-trivial task due to the UTXO architecture of Corda.

Potentiality Assessment (PA)	**Score (0-4)**
P1: Interoperation within the same DLT network, same subnetworks	☐
P2: Interoperation within the same DLT network, different subnetworks	☐
P3: Interoperation within different DLT networks	☐
P4: Interoperation within different DLT protocols	☐
Compatibility Assessment (CA)	**Score (0-3)**
C1: Provides semantic-level interoperability (shared protocols)	☐
C2: Provides organization-level interoperability (shared agreements)	☐
C3: Provides legal-level interoperability (follow regulations)	☐
Performance Assessment (PeA)	**Score (0-3)**
PE1: Provides acceptable cross-chain transaction end-to-end latency/throughput	☐
PE2: Provides acceptable cross-chain transaction end-to-end cost	☐
PE3: Complies with desirable energetic consumption goals	☐
PA + CA + PeA	**Total (0-10):**

Table 3. DLT interoperability solution assessment. Interoperability assessment is divided into PE, CA, and PeA assessments. A higher score corresponds to a more interoperable solution.

- *Level P2*: interoperation within the same DLT network, different subnetworks (smart contracts can call other smart contracts from other subnetworks, e.g., across Hyperledger Fabric channels).
- *Level P3*: interoperation across DLT networks of the same DLT protocol, i.e., homogeneous blockchains (e.g., Ethereum Ropsten to Ethereum mainnet, Hyperledger Fabric network A to Hyperledger Fabric network B, as seen in [52]).
- *Level P4* interoperation across different networks of different DLT protocols, i.e., heterogeneous DLTs, as seen in [52, 93, 113].

We could also consider *Level P5*, providing interoperation with non-DLT systems (e.g., enterprise systems, payment systems). However, all IM solutions and DLT nodes provide capabilities for accessing the ledger. In the compatibility assessment, we consider interactions (e.g., digital asset exchanges across DLTs) to have legal binding.

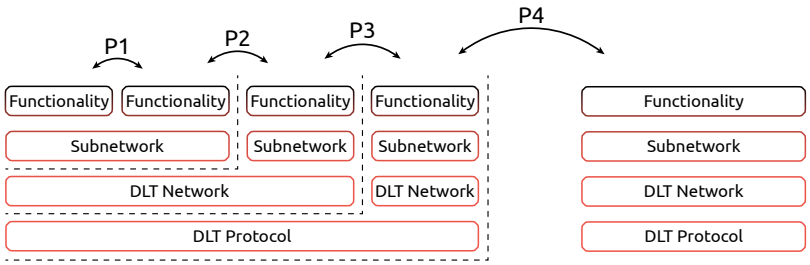


Fig. 11. Potentiality assessment of an IM

For instance, Hyperledger Fabric-based networks can provide and receive information from and to the exterior, respectively, via smart contracts. Figure 12 depicts a practical example of a potentiality assessment. A high score for this

assessment shows that *the system can interoperate with systems significantly different from it as is*. One could consider an IM as the cross-chain logic plus a connection mode (in this case a DLT gateway). The IM can then connect to multiple DLT networks, depending on how many of the latter the DLT gateway supports. Thus, business logic from the IM can spawn across several DLT networks. If this is the case, cross-chain logic can be implemented. Another interesting possibility can be implemented: connecting the IMs via the DLT gateways, allowing for second-order interoperability. Provided accountability guarantees, such as smart contracts as trust anchors, or a decentralized log storage for IMs, this enables cross-chain use cases operated by mutually untrusted IMs

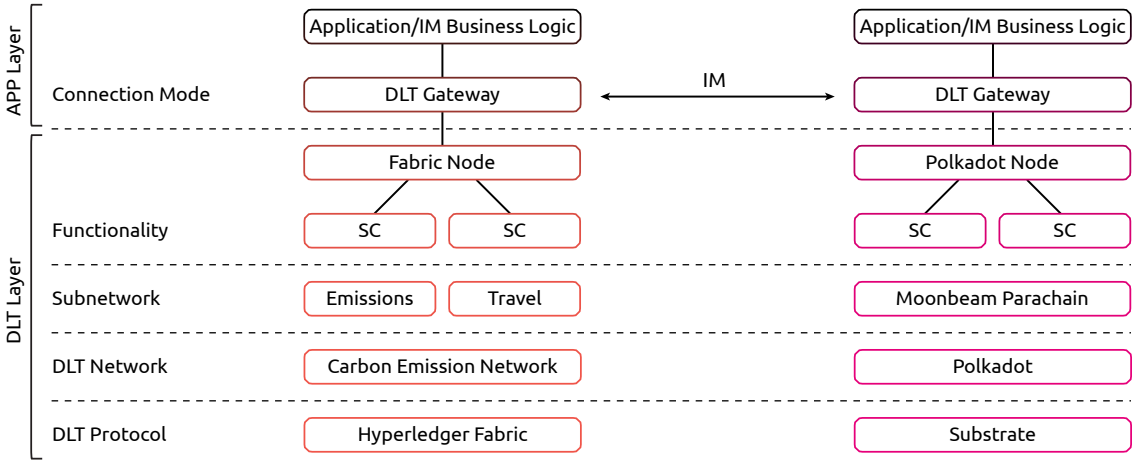


Fig. 12. Example of vertical interoperation in a Hyperledger Fabric network and the Polkadot network. Horizontal interoperability can be achieved via an IM using, for example, a DLT gateway.

4.1.2 Compatibility assessment. The compatibility assessment evaluates compatibility aspects regarding semantic, organizational, and legal interoperability. *Given a pair of systems, do they run protocols that both understand? Do they share similar organizational goals? Do they follow the same jurisdiction and regulations?*

Figure 13 depicts this assessment. Three levels of compatibility maturity exist:

- *Level C1*: semantic interoperability is achieved by a pair of systems (see Section 2)
- *Level C2:* semantic and organizational interoperability are achieved
- *Level C3:* semantic, organizational, and legal interoperability is achieved.

The score for this assessment is obtained by summing the weights of each level cumulatively, since the last layer typically depends on the previous (1, 2, and 3, respectively). In this paper, we focus on the semantic aspect, leaving pointers for future work on the organizational and legal aspects. The granularity regarding the three layers can be defined by the users of the framework.

Figure 14 depicts an architecture of an interoperability solution (e.g., it could be connected by a network of gateways) that has a compatibility assessment conducted. A network of gateways is a set of gateways that run cross-chain logic shared by gateways, following a protocol.

4.1.3 Performance assessment. The performance assessment studies how efficiently an IM executes its processes. The efficiency can be measured in metrics related to the Cross-chain transaction (CC-Tx) concept. A CC-Tx is composed of

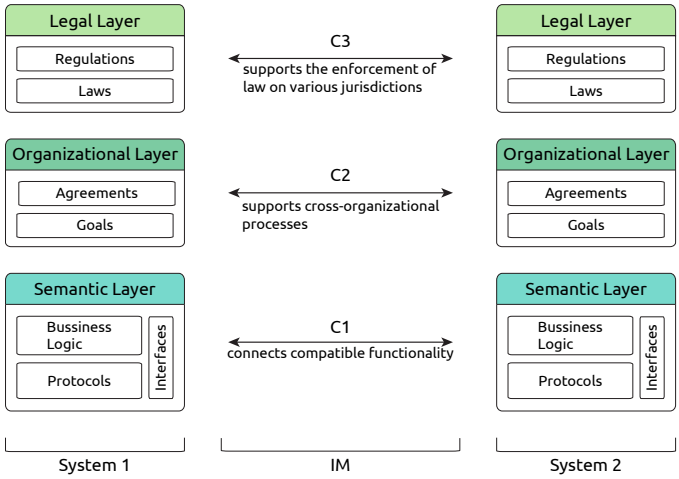


Fig. 13. Compatibility assessment between two systems

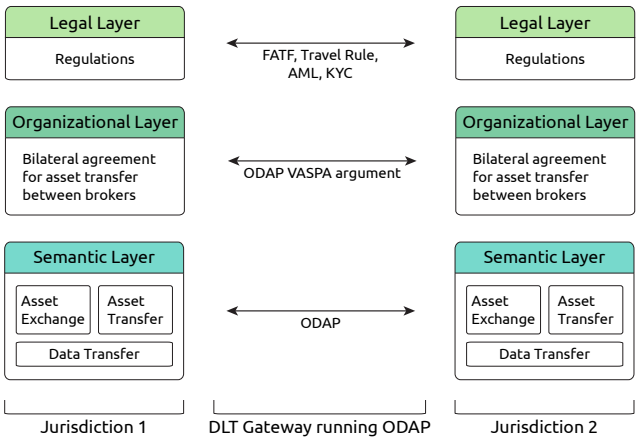


Fig. 14. Example of interoperation between two DLTs connected by a DLT gateway network running a digital asset transfer protocol such as ODAP [61]. A compatibility assessment can be performed regarding the participating DLTs.

transactions directed to the target systems (called *subtransactions of a CC-Tx*) plus the internal transactions of the IM. We call the logic that an IM executes *cross-chain logic* or *cross-chain rules*. The cross-chain logic accounts for business logic plugins' execution, which can modify the final transactions issued against DLTs.

Following the blockchain integration framework, there are three main metrics to assess the performance of an IM [91]:

- *End-to-end latency*: calculated by summing the time to execute the IM cross-chain logic plus the latency of every sub-transaction (until it is committed and finalized, in its local DLT). For example, the Carbon Emission use case (see Figure 3) would have a latency calculated by the execution of the mdApp logic plus two transactions on the

Emissions Channel plus one transaction on the Ethereum network. The end-to-end latency answers the question *how long does it take for the cross-chain transaction to be incorporated on the target systems*?

- *End-to-end throughput*: the throughput is the number of cross-chain transactions executed per second. The more complex the cross-chain logic, and the longer it takes for each sub-transaction to be committed on their target system, the lower the throughput. The throughput metric answers the question *how many cross-chain transactions are finalized on the ledgers per second?*
- *End-to-end cost*: cost can come in two forms: transaction cost and energetic consumption. Cost can be measured in transaction fees and/or direct transaction costs. Transaction fees are paid to support the network (happening in public, permissionless DLTs more often). Direct transaction cost (in a certain period) can be calculated by dividing the cost of being in the network by the number of transactions. This latter model is usual on permissioned networks with the subscription business model. The energetic consumption is calculated by dividing the energetic cost of a transaction per number of transactions. This metric answers the question *what is the cost in transaction fees and energetic consumption of the sum of the cross-chain transactions issued by the system?*.

At the moment, it is not possible to establish specific guidelines for this type of assessment due to the lack of systematic evaluations of interoperability solutions. Although several solutions bring performance evaluations [12], they are not standardized according to the any framework, making it difficult, if not impossible, to compare solutions systematically. Thus, we leave the judgment of a reasonable latency, throughput, and cost for interoperability solutions and a rigorous model to evaluate the performance for future work. We emphasize the challenges to measuring the energetic consumption of an IM, given that it interacts will multiple decentralized systems. Although some work has been done in evaluating the energetic consumption of Bitcoin [54, 56], the literature falls short in exploring other DLTs. Thus, this remains a problematic metric to assess.

4.2 Choosing the right interoperability solution

In this section, we help the reader choose an IM, using two decision models. The proposed decision models are directed to researchers,developers, and software architects. The first decision model focuses on assisting the choice of the IM's infrastructure (connection mode), for a given use case. The second decision model assists in the choice of the IM's functionality (interoperation mode, potentiality, and compatibility). The output of functionality diagram suggests a group of IM for the chosen functionality.

How to use the decision models. The reader should start navigating from left to right, starting on the node with the *START* label. After that, a path should be followed by answering yes (✓) or (✗) to the proposed questions until an end node is reached (blue nodes), or a *proceed* flag is present in one of the arrows connect to the current node. More details on the proceed flag are available on the functionality decision model. The blue nodes output recommendations regarding the infrastructure or functionality of an IM.

Infrastructure. Choosing an infrastructure refers to choosing the hosting infrastructure for a IM: DLT node, DLT proxy, or DLT gateway. Hosting an IM implies several challenges that require specialized, well-trained experts: 1) node and hardware management, including installation, maintenance, load balancing, software version management, redundancy, and scaling; 2) security, including monitoring the node and responding to cyber-attacks; 3) and others, such as adhering to regulations (e.g., GDPR). Thus, different needs require a different infrastructure. Depending on the use case, it is acceptable to defer the management of the infrastructure to third parties (versus self-hosting the infrastructure).

The following decision model, in Figure 15, guides on choosing the infrastructure (i.e., connection mode) for an IM and if it should be self-hosted or not.

DLT nodes are native blockchain clients, e.g., Geth Ethereum Node [43], Hyperledger Besu node (Ethereum node) [5], Hyperledger Fabric peer node [5], Bitcoin Core [14], and Polkadot node [108]. DLT proxies include Infura [68], Blockdaemon [15], and nodes hosted and accessible via cloud providers. DLT gateways include Quant Overledger [113], Hyperledger Cactus nodes [93], Weaver gateways [127], among others. More examples of each connection mode can be found in [12].

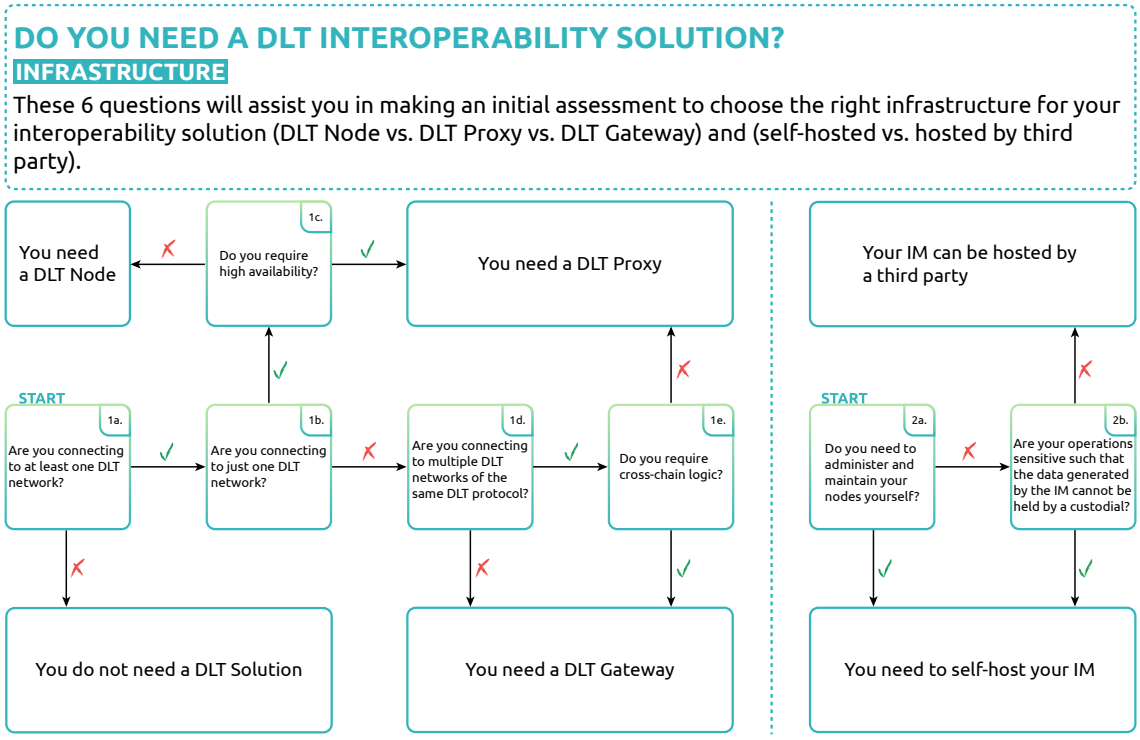


Fig. 15. Decision map guiding the choice of the infrastructure of a DLT interoperability solution. Start on the top most left node (with the green *START* label) and answer the questions until you arrive to an end node (blue nodes). The output of this process is a group of IM that respect the requirements stated on the process flow.

Functionality. Choosing the functionality refers to choosing the IM functionality in terms of interoperation mode, P levels, and C levels.

Most IM assure P $\leq$ 3, while a few provide all P-levels (P1, P2, P3, and P4) [12]. On the other hand, IMs can be divided on C1, because most do provide $C < 2$. Most IMs can provide C1 (in fact, a system providing P4 implies that it provides C1), while a few attempts at implementing standards that could, in the future, support the legal layer. At the moment, we do not know of any IM providing level C3.

The following decision model, in Figure 16, guides on choosing the functionality. This decision model uses the *proceed* flag, meaning that when it is present in an arrow connected to the current node, the user should evaluate the

condition, and then proceed (instead of stopping), *accumulating* the recommendations, until a node without a proceed node is found (and therefore the last decision is made on that node). Take, for example, the following flow: one starts in node *3a*. As the *proceed* flag is present on that node, the reader will answer to the question and then (independently of the answer) move to the next node. In case the answer was yes (✓), the recommendation is saved. At the end of Figure 16, a maximum of five recommendations may be collected.

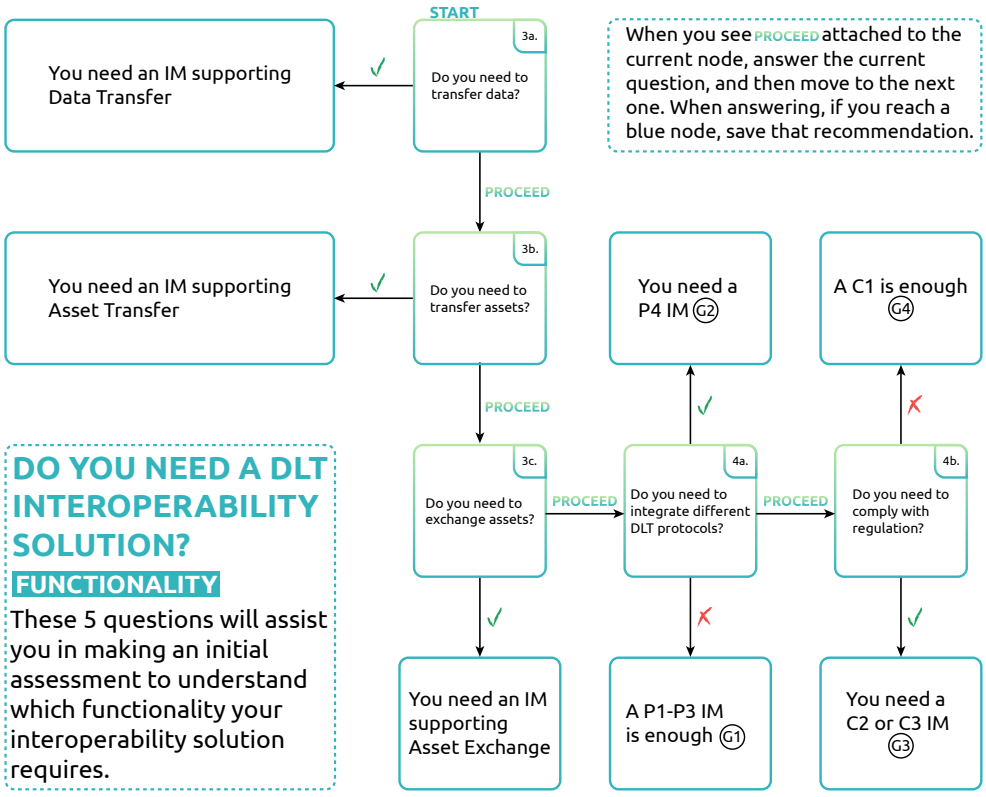


Fig. 16. Decision map guiding the choice of the functionality of a DLT interoperability solution. Start on the topmost left node (with the green label START) and answer the questions until you arrive to an end node (grey ones). The output of this process is a group of IM that respect the requirements stated on the process flow.

4.2.1 Solution Groups. In this section, we define each solution group depicted in Figure 16. In particular, we systematically compare IMs according to their P level, C level, and interoperation mode. Table 4 shows examples of solutions belonging to each group proposed by Figure 16.

The first group comprises solutions providing levels P1-P3, and supporting data transfers, asset transfers, or asset exchanges. Most blockchains provide P1 interoperability by enabling functionality re-usage. For instance, smart contracts can call other smart contracts (even across subnetworks, providing level P2) in most blockchains. Level P2 requires some orchestration. For example, interoperability across subnetworks in Ethereum can be done via oracles or gateways running bespoke cross-chain logic. A similar level of orchestration happens when P3 is needed but could be more complex, as different networks may differ more than different subnetworks. Level P3 and P4 systems imply the

Solution Group	Solution	P1	P2	P3	P4	C1	C2	C3	$\mathcal{D}$	$\mathcal{A}_t$	$\mathcal{A}_e$
G1	[2, 22, 39, 104, 117]	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗
	[1, 4, 41, 99, 112, 116], HTLCs [12]	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓
	Blockchain of blockchains (e.g., [79, 129])	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓
G2	[47, 48, 52, 93, 100, 127]	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗
	[11, 17, 32, 33, 65, 66, 70, 93, 138], HTLCs [12]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
	[93, 109, 117], Bridges (e.g., [7, 82, 83])	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓
G3	Solutions supporting P $\geq$ 3										
G4	[6, 113]	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓

Table 4. Categorization of IM according to P levels, C levels, and interoperation mode ($\mathcal{D}$ stands for data transfer, $\mathcal{A}_t$ for asset transfer, and $\mathcal{A}_e$ for asset exchange).

usage of blockchain interoperability middleware (in some DLT protocols, level P3 can be achieved without specialized middleware) that can be centralized or decentralized.

Many of the examples present in group G2 supporting data transfers are classified as *trusted relays* or *blockchain agnostic protocols*, while G2 solutions supporting asset transfers and exchanges belong to *sidechains & relays*. Blockchain of blockchain platforms (Polkadot, Cosmos) provides level P3. The design and implementation of bridges to external DLTs would carry these systems to level P4, but they are still in development, and, furthermore, these bridges are not native to the DLT networks.

Example 1: oracle solution. In this section, we present an example of the choice of the infrastructure and functionality of an oracle solution based on a simple use case. Let us consider an IM administrator who transfers data between the Ethereum blockchain and a Polkadot's parachain.

Infrastructure: the administrator does not want to host the infrastructure and would like to have high availability. There is no preference to whether the IM can access the DLT nodes directly or not. The administrator accepts the risk of its data being read by third parties, for the comfort of an easier to deploy solution. Using the decision model from Figure 15, the administrator would answer:

(1) **What is your desired connection mode?**
- 1a: Are you connecting to at least one DLT? ✓
- 1b: Are you connecting to just one DLT? ✗
- 1c: Do you need high availability? ✓
- 1d: Do you need to run cross-chain logic? ✓

Therefore, a DLT gateway is needed.

(2) **What is your desired hosting mode?**
- 2a: Do you need to administer and maintain your nodes yourself? ✗

- 2b: Are your operations sensitive such that the data generated by the IM cannot be held by a custodial? ✗

Therefore, the IM can be hosted by a third party.

Functionality: the IM would only need to transfer data. It needs to connect different DLT protocols and does not need to comply with any regulations. Using the decision model from Figure 16, the administrator would answer:

(1) **What is the desired interoperation mode?**
- 3a: Do you need to transfer data? ✓
- 3b: Do you need to transfer assets? ✗
- 3c: Do you need to exchange assets? ✗

Therefore, IM supporting asset transfers is needed.

(2) **What is the desired supported functionality?**
- 4a: Do you need to integrate different DLT protocols? ✓
- 4b: Do you need to comply with regulation? ✗

Therefore, solutions from G2 (supporting P1-P4, C1, and data transfers) are needed.

Therefore, the chosen IM is a P4 + C1 third-party hosted DLT gateway (see Table 4 to choose a solution).

Example 2: cross-authentication solution. In this section, we present an example of the choice of the infrastructure and functionality of an cross-authentication solution based on a simple use case. Let us consider an end-user who wants to perform an asset exchange. The user wants to exchange asset a in DLT a for asset b in DLT b.

Infrastructure: the user is looking to connect to two different DLTs, does not need to run cross-chain logic (as the only logic needed is a time-locked transfer), with no high availability requirements. The user wants to have direct access and control over the node. Using the decision model from Figure 15, the administrator would answer:

(1) **What is your desired connection mode?**
- 1a: Are you connecting to at least one DLT? ✓
- 1b: Are you connecting to just one DLT? ✗
- 1c: Do you need high availability? ✗
- 1d: Do you need to run cross-chain logic? ✗

Therefore, a DLT proxy (or DLT node) is needed.

(2) **What is your desired hosting mode?**
- 2a: Do you need to administer and maintain your nodes yourself ✓
- 2b: Are your operations sensitive such that the data generated by the IM cannot be held by a custodial? ✗

Therefore, the IM can be self hosted.

Functionality: the IM needs to exchange assets (or perform two independent asset transfers). It needs to connect different DLT protocols and do not need to comply with any regulation, but it must comply with rules on exchanging assets (e.g., hashlock time contract). Using the decision model from Figure 16, the administrator would answer:

(1) **What is the desired interoperation mode?**

- 3a: Do you need to transfer data? ✗
- 3b: Do you need to transfer assets? ✗
- 3c: Do you need to exchange assets? ✓

Therefore, an IM supporting asset exchanges is needed.

(2) **What is the desired supported functionality?**

- 4a: Do you need to integrate different DLT protocols? ✓
- 4b: Do you need to comply with regulation? ✗

Therefore, we need solutions from G2 (supporting P1-P4, C1, and supporting asset exchanges).

Therefore, the chosen IM is a P4 + C1 self-hosted DLT proxy supporting asset exchanges.

5 RELATED WORK AND OPEN RESEARCH CHALLENGES

In this section, we compare the contributions of our paper to the state-of-the-art.

Status quo of Blockchain interoperability. Compared to the existing literature, our study focuses on allowing researchers and developers (or software architects) to choose a blockchain interoperability solution, which was only partially addressed before. In particular, a number of surveys studied blockchain interoperability solutions or architectures [12, 13, 16, 24, 73, 74, 77, 88, 111, 119, 120, 126, 137], but did not provide a decision model to choose one. Each survey comes with trade-offs (technical explanation depth vs. IM coverage) and often with conflicting categories of solutions. This implies that the reader will not have a holistic overview of the area. Our survey performs an analysis of these surveys, attempting to unify and synthesize existing knowledge.

The latest systematic survey on IM is Belchior et al.'s [12]. In this survey, the authors categorize IM into public connectors, hybrid connectors, and blockchain of blockchains. Instead of answering *how* are IM connecting DLTs, in the survey, we classify IM according to *what they connect*.

Our categorization departs from older ones because we consider trusted relays, sidechains, notary schemes, relays, gateways, and other design patterns built on top of oracles. We clarify that only two types of interoperability exist and propose a general IM model.

Assessing the degree of interoperability of an IM. There is extensive work in the area of interoperability assessment. Leal et al. discuss and systematically compare 21 interoperability assessment approaches [36]. Most approaches aim to assess the interoperability of (centralized) systems in the technical, semantic, and organizational layers, while some generically evaluate interoperability.

In [125], the authors discuss different interoperability testing architectures for assessing interoperability between distributed systems. The authors provide guidelines to generate interoperability tests for the proposed testing architectures. Our work provides guidelines (or interoperability tests) for DLTs. Numerous other works evaluate interoperability for IoT [139], cloud providers [75, 84, 121], and more generic ones [76, 81].

The closest to our work is Mihaiu et al.'s *blockchain interoperability evaluation framework* [91]. In this study, the authors introduce the concepts of cross-chain rules and propose a list of metrics to compare blockchain interoperability solutions. In an older study, the authors compare two interoperability solutions based on twelve ad-hoc criteria [77].

To the best of our knowledge, our framework is the first to provide insights on the potentiality, compatibility, and performance of a blockchain interoperability solution while explicitly providing support to choose the infrastructure and functionality of an IM. Our framework helps the reader confirm the IM's adequacy for interoperating with other systems at technical, semantic, organizational, and legal levels. Our framework for choosing a blockchain interoperability solution may resemble studies aimed at facilitating the choice of an IM on functional and non-functional requirements, such as [135].

Open Research Challenges. Although recent years have assisted to a skyrocketing increase in cross-chain research, several research challenges are left unsolved: First and foremost, there is no formalization of a general model for IMs (using frameworks to define and prove security properties such as the universal composability framework [26]) that can answer the questions: *what are the technical requirements that a DLT must provide, in order to be interoperable?*, *what are the technical requirements an IM must provide to assure safety and liveness properties?*. These research questions are important because often implementations of IM typically do not follow specific guidelines [25].

Secondly, as pointed out by [8, 12] there is still a lack of supporting tools for IMs, such as monitoring tools (e.g., visualization and analysis of cross-chain transactions and state), cross-chain digital identity, migration tools (change the DLT infrastructure on the go for a dApp or mDApp), security tools (automatic detection of frontrunning attacks, formal analysis of cross-chain protocols), and others.

Following the reasoning line of this work, there are still no methods to systematically evaluate and compare the performance of an IM. Although we propose an initial set of metrics to respond to this need, there are no baseline benchmarks.

Finally, regulation on cross-chain asset transfers, mainly amongst institutional players, will significantly impact how modern financial systems interact and evolve. Although there have been recent and numerous efforts on regulating certain aspects of interoperability (such as the ODAP protocol [61]), currently, there are no available standards.

Key Takeway 7. *DLT interoperability standards*

Although none are official standards, there is a wide range of standardization efforts on blockchain interoperability. The works in [12, 102, 131] survey the major standardization efforts in the field.

6 CONCLUSION

The future of DLT technology depends on its capability to interoperate across different dimensions. To address such a gap, interoperability solutions flourished in recent years. However, several challenges are posed to researchers and practitioners when trying to understand blockchain interoperability. For instance, there is still no systematic way for classifying, assessing, comparing, and choosing DLT interoperability solutions.

Our paper closes this gap by exposing three main contributions. First, we propose a unified conceptual model and classification framework for blockchain interoperability solutions. Second, we propose a framework to assess the interoperability capabilities of a system utilizing one or more DLTs. Lastly, we propose two decision models that allow one to choose the infrastructure and functionality for an IM, to enable their off-chain systems to interoperate with multiple DLTs. We provide practical examples of this decision process. Finally, based on the most recent research, we provide a list of updated open research challenges in the blockchain interoperability research area.

ACKNOWLEDGMENTS

We warmly thank our colleagues in the IETF's forming working group ODAP for fruitful discussions. We thank the Hyperledger Cactus community and Iulia Mihaiu for insightful discussions on blockchain interoperability. This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID), and 2020.06837.BD. Rafael was supported by Quant.

REFERENCES

[1] 0x Protocol Team. 0x: Powering the decentralized exchange of tokens on Ethereum, 2021. Available online: https://0x.org/, last accessed on 2022-01-10.

[2] Abebe, E., Behl, D., Govindarajan, C., Hu, Y., Karunamoorthy, D., Novotny, P., Pandit, V., Ramakrishna, V., and Vecchiola, C. Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer. In *Proceedings of the 20th International Middleware Conference Industrial Track* (2019), Association for Computing Machinery, pp. 29–35.

[3] Abebe, E., Karunamoorthy, D., Yu, J., Hu, Y., Pandit, V., Irvin, A., and Ramakrishna, V. Verifiable Observation of Permissioned Ledgers. *arXiv 2012.07339v2* (2021). Available online: https://arxiv.org/abs/2012.07339, last accessed on 2022-01-10.

[4] Adams, H., Zinsmeister, N., and Robinson, D. Uniswap v2 Core. Tech. rep., 2020. Available online: https://docs.uniswap.org/, last accessed on 2022-01-10.

[5] Androulaki, E., Barger, A., Bortnikov, V., Muralidharan, S., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Murthy, C., Ferris, C., Laventman, G., Manevich, Y., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., and Yellick, J. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the 13th EuroSys Conference, EuroSys 2018* (New York, New York, USA, 4 2018), vol. 2018-Janua, Association for Computing Machinery, Inc, pp. 1–15.

[6] ARK Team. ARK Whitepaper Version 2.1.0, 2019. Available online: https://whitepaper.ark.io/prologue, last accessed on 2022-01-10.

[7] Baneth, T. Waterloo — a Decentralized Practical Bridge between EOS and Ethereum, 2019. Available online: https://blog.kyber.network/waterloo-a-decentralized-practical-bridge-between-eos-and-ethereum-1c230ac65524, last accessed on 2021-02-19.

[8] Belchior, R. PhD Thesis Proposal - Blockchain Interoperability. Tech. rep., Instituto Superior Técnico, 9 2021. Available online: https://www.researchgate.net/publication/355370486_PhD_Thesis_Proposal, last accessed on 2021-08-10.

[9] Belchior, R., Guerreiro, S., Vasconcelos, A., and Correia, M. A Survey on Business Process View Integration. To appear: Business Process Management Journal.

[10] Belchior, R., Vasconcelos, A., Correia, M., and Hardjono, T. Enabling Cross-Jurisdiction Digital Asset Transfer. In *IEEE International Conference on Services Computing* (2021), IEEE.

[11] Belchior, R., Vasconcelos, A., Correia, M., and Hardjono, T. HERMES: Fault-Tolerant Middleware for Blockchain Interoperability. *Future Generation Computer Systems* (3 2021).

[12] Belchior, R., Vasconcelos, A., Guerreiro, S., and Correia, M. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. *ACM Computing Surveys* (5 2021).

[13] Bishnoi, M., and Bhatia, R. Interoperability solutions for blockchain. *Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020* (2020), 381–385.

[14] Bitcoin. Bitcoin Core integration/staging tree, 2021. Available online: https://github.com/bitcoin/bitcoin, last accessed on 2021-10-14.

[15] Blockdaemon. Platform - Blockdaemon, 2021. Available online: https://blockdaemon.com/platform/, last accessed on 2022-01-10.

[16] Borkowski, M., Frauenthaler, P., Sigwart, M., Hukkinen, T., Hladky, O., and Schulte, S. Cross-Blockchain Technologies: Review, State of the Art, and Outlook, 2019. Available online: https://dsg.tuwien.ac.at/projects/tast/pub/tast-white-paper-4.pdf, last accessed on 2021-08-10.

[17] Borkowski, M., Sigwart, M., Frauenthaler, P., Hukkinen, T., and Schulte, S. DeXTT: Deterministic Cross-Blockchain Token Transfers. *IEEE Access 7* (8 2019), 111030–111042.

[18] Breidenbach, L., Cachin, C., Chan, B., Coventry, A., Ellis, S., Juels, A., Koushanfar, F., Miller, A., Magauran, B., Moroz, D., Nazarov, S., Topliceanu, A., Tramèr, F., and Zhang, F. Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks. Tech. rep., 2021. Available online: https://research.chain.link/whitepaper-v2.pdf, last accessed on 2021-08-10.

[19] Briefing, C. Aave is Exploring Solana, Avalanche, Layer 2 Expansion - Crypto Briefing, 2021. Available online: https://cryptobriefing.com/aave-is-exploring-solana-avalanche-layer-2-expansion/, last accessed on 2022-01-10.

[20] Brown, R. The Corda Platform: An Introduction White Paper, 2018. Available online: https://www.r3.com/reports/the-corda-platform-an-introduction-whitepaper, last accessed on 2021-08-10.

[21] BSC, M. Binance Launches $1B Binance Smart Chain Fund to Reach One Billion Crypto Users, 2021. Available online: https://www.binance.org/en/blog/binance-launches-one-billion-binance-smart-chain-fund-to-reach-one-billion-crypto-users/, last accessed on 2022-01-10.

[22] Bu, G., Haouara, R., Nguyen, T. S. L., and Potop-Butucaru, M. Cross hyperledger fabric transactions. In *CRYBLOCK 2020 - Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Part of MobiCom 2020* (2020), Association for Computing Machinery, pp. 35–40.

[23] Buterin, V. Ethereum White Paper, 2009. Available online: https://www.networkworld.com/article/2177684/lan-wan/the-growth-in-east-west-traffic.html, last accessed on 2022-01-10.

[24] Buterin, V. R3 Report - Chain Interoperability. Tech. rep., R3 Corda, 2016. Available online: https://www.r3.com/wp-content/uploads/2017/06/chain_interoperability_r3.pdf, last accessed on 2021-08-10.

[25] Caldarelli, G., and Ellul, J. The blockchain oracle problem in decentralized finance—A multivocal approach. *Applied Sciences (Switzerland) 11*, 16 (2021).

[26] Canetti, R. Universally composable security: A new paradigm for cryptographic protocols. *Annual Symposium on Foundations of Computer Science - Proceedings* (2001), 136–145.

[27] Carbon Accounting and Certification WG. Carbon Accounting and Certification WG - Climate Action SIG - Hyperledger Foundation, 2021. Available online: https://wiki.hyperledger.org/display/CASIG/Carbon+Accounting+and+Certification+WG, last accessed on 2022-01-10.

[28] Carbon Emission Working Group. Hyperledger Working Groups - Blockchain Carbon Accounting, 2021. Available online: https://github.com/hyperledger-labs/blockchain-carbon-accounting, last accessed on 2022-01-10.

[29] Chalyvidis, C. E., Ogden, J. A., Johnson, A. W., Colombi, J. M., and Ford, T. C. A method for measuring supply chain interoperability. 246–258. Available online: https://www.tandfonline.com/doi/abs/10.1080/16258312.2016.1247655, last accessed on 2021-08-10.

[30] Chen, D., Doumeingts, G., and Vernadat, F. Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry 59*, 7 (9 2008), 647–659.

[31] Coin Market Cap. All Coins, 2021. Available online: https://coinmarketcap.com/coins/views/all, last accessed on 2021-08-10.

[32] Composable Finance. Composable Finance Whitepaper, 2021. Available online: https://paper.composable.finance, last accessed on 2021-12-21.

[33] Connext. Welcome! | Connext Documentation, 2021. Available online: https://docs.connext.network, last accessed on 2021-08-10.

[34] Consensys. DeFi Report Q2 2021 | Consensys. Available online: https://consensys.net/reports/defi-report-q2-2021, last accessed on 2021-08-12.

[35] Correia, M. From Byzantine Consensus to Blockchain Consensus. *Essentials of Blockchain Technology* (2019), 41.

[36] da Silva Serapião Leal, G., Guédria, W., and Panetto, H. Interoperability assessment: A systematic literature review. *Computers in Industry 106* (4 2019), 111–132.

[37] da Silva Serapião Leal, G., Guédria, W., and Panetto, H. Interoperability assessment: A systematic literature review. *Computers in Industry 106* (4 2019), 111–132.

[38] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. *Proceedings - IEEE Symposium on Security and Privacy 2020-May* (5 2020), 1106–1120.

[39] DAML. Canton, 2021. Available online: https://www.canton.io, last accessed on 2021-08-10.

[40] Dhillon, A., McBurney, P., Kotsialou, G., and Riley, L. Voting over a distributed ledger: An interdisciplinary perspective. *SocArXiv* (2020).

[41] DYdX. dYdX, 2021. Available online: https://dydx.exchange, last accessed on 2021-08-10.

[42] Eskandari, S., Salehi, M., Catherine Gu, W., and Clark, J. SoK: Oracles from the Ground Truth to Market Manipulation; SoK: Oracles from the Ground Truth to Market Manipulation. *arxiv 2106.00667* (2021).

[43] Ethereum Foundation. Getting Started with Geth | Go Ethereum, 2021. Available online: https://geth.ethereum.org/docs/getting-started, last accessed on 2021-12-17.

[44] Ethereum Foundation, and Consensys. BTC-relay: Ethereum contract for Bitcoin SPV, 2015. Available online: https://github.com/ethereum/btcrelay, last accessed on 2020-03-20.

[45] European Commission. NIFO - National Interoperability Framework Observatory, Interoperability layers, 2021. Available online: https://joinup.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory/3-interoperability-layers, last accessed on 2020-04-20.

[46] Extropy.io. Price Oracle Manipulation | by Extropy.IO | Sep, 2021 | Medium, 2021. Available online: https://extropy-io.medium.com/price-oracle-manipulation-d46fd413cc17, last accessed on 2021-09-17.

[47] Falazi, G., Breitenbücher, U., Daniel, F., Lamparelli, A., Leymann, F., and Yussupov, V. Smart Contract Invocation Protocol (SCIP): A Protocol forthe Uniform Integration of Heterogeneous Blockchain Smart Contracts. *CAiSE 2020 1* (2020), 134–149.

[48] Frauenthaler, P., Sigwart, M., Spanring, C., Sober, M., and Schulte, S. ETH Relay: A Cost-efficient Relay for Ethereum-based Blockchains. *Proceedings - 2020 IEEE International Conference on Blockchain, Blockchain 2020* (11 2020), 204–213.

[49] Garay, J., Kiayias, A., and Leonardos, N. The Bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology* (2015), vol. 9057, pp. 281–310.

[50] Garoffolo, A., Global Horizen, A., Kaidalov, D., and Oliynykov, R. Zendoo: a zk-SNARK Verifiable Cross-Chain Transfer Protocol Enabling Decoupled and Decentralized Sidechains. *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)* (2020).

[51] Gartner. Blockchain Platforms Reviews 2021 | Gartner Peer Insights, 2021. Available online: https://www.gartner.com/reviews/market/blockchain-platforms, last accessed on 2021-07-22.

[52] Ghaemi, S., Rouhani, S., Belchior, R., Cruz, R. S., Khazaei, H., and Musilek, P. A Pub-Sub Architecture to Promote Blockchain Interoperability. *Submitted to Computer Communications* (1 2021). Preprint available online: https://deepai.org/publication/a-pub-sub-architecture-to-promote-blockchain-interoperability, last accessed on 2021-08-10.

[53] Giulio, C. Wrapping trust for interoperability. A study of wrapped tokens. Tech. rep., 2021. Preprint available online: https://arxiv.org/pdf/2109.06847v1.pdf, last accessed on 2021-08-10.

[54] Giungato, P., Rana, R., Tarabella, A., and Tricase, C. Current Trends in Sustainability of Bitcoins and Related Blockchain Technology.

Sustainability 2017, Vol. 9, Page 2214 9, 12 (11 2017), 2214.

[55] Gkritsi, E., and Shen, M. Cross-Chain DeFi Site Poly Network Hacked; Hundreds of Millions Potentially Lost | Nasdaq. Available online: https://www.nasdaq.com/articles/cross-chain-defi-site-poly-network-hacked-hundreds-of-millions-potentially-lost-2021-08-10, last accessed on 2021-08-10.

[56] Gonzalez-Barahona, J. M. Factors determining maximum energy consumption of Bitcoin miners. Available online: https://digiconomist.net/bitcoin-energy-consumption, last accessed on 2021-08-10.

[57] Gottschalk, P. Maturity levels for interoperability in digital government. *Government Information Quarterly 26*, 1 (1 2009), 75–81.

[58] Grants, B. Defi & NFT Blockchain Grants. Available online: https://www.blockchaingrants.org/home, last accessed on 2021-08-10.

[59] Hardjono, T., Hargreaves, M., Smith, N., and Ramakrishna, V. An Interoperability Architecture for Blockchain Gateways. Internet-draft draft-hardjono-blockchain-interop-arch-03, IETF, November 2021.

[60] Hargreaves, M., and Hardjono, T. Implementing a CBDC: the challenges and a solution. Tech. rep., 2021. Available online: https://www.quant.network/insights/implementing-a-cbdc-the-challenges-and-a-solution, last accessed on 2022-01-10.

[61] Hargreaves, M., Hardjono, T., and Belchior, R. Open Digital Asset Protocol draft 02. Tech. Rep. draft-hargreaves-odap-02, Internet Engineering Task Force, 2021. Available online: https://datatracker.ietf.org/doc/html/draft-hargreaves-odap-02, last accessed on 2022-01t-10.

[62] Heiler, S. Semantic Interoperability. *ACM Computing Surveys (CSUR)* (1995).

[63] Henninger, A., and Mashatan, A. Distributed Interoperable Records: The Key to Better Supply Chain Management. *Computers 2021, Vol. 10, Page 89 10*, 7 (7 2021), 89.

[64] Herlihy, M., Liskov, B., and Shrira, L. Cross-chain Deals and Adversarial Commerce. *Very Large Databases 13*, 2 (2019), 100–113.

[65] Hop Exchange. Hop Exchange, 2021. Available online: https://kovan.hop.exchange/send, last accessed on 2022-01-10.

[66] Hyphen. Hyphen - Instant Cross-Chain Transfers - Biconomy, 2021. Available online: https://docs.biconomy.io/products/hyphen-instant-cross-chain-transfers?ref=hackernoon.com, last accessed on 2022-01-10.

[67] Ide, N., and Pustejovsky, J. What Does Interoperability Mean, Anyway? Toward an Operational Definition of Interoperability for Language Technology. *Conference on Global Interoperability* (2010).

[68] Infura. Ethereum | Infura Documentation, 2021. Available online: https://infura.io/docs/ethereum, last accessed on 2022-01-10.

[69] Interchain Foundation. Funding | Interchain Foundation. Available online: https://interchain.io/funding/, last accessed on 2022-01-10.

[70] Interledger. Interledger Protocol V4 (ILPv4) | Interledger, 2020. Available online: https://interledger.org/rfcs/0027-interledger-protocol-4, last accessed on 2022-01-10.

[71] ISO. Requirements for establishing manufacturing enterprise process interoperability — Part 1: Framework for enterprise interoperability. Available online: https://www.iso.org/standard/50417.html, last accessed on 2021-07-22.

[72] ISO - TC 307. Blockchain and distributed ledger technologies — Vocabulary (Draft ISO/TC 307/WG 1 N 783 ISO/TC). Tech. rep., 2020. Available online: https://www.iso.org/standard/73771.html, last accessed on 2022-01-10.

[73] Johnson, S., Robinson, P., and Brainard, J. Sidechains and interoperability, 3 2019. Available online: http://arxiv.org/abs/1903.04077, last accessed on 2022-01-10.

[74] Kannengiesser, N., Pfister, M., Greulich, M., Lins, S., and Sunyaev, A. Bridges Between Islands: Cross-Chain Technology for Distributed Ledger Technology. *Hawaii International Conference on System Sciences* (2020).

[75] Kaur, K., Sharma, S., and Kahlon, K. S. Interoperability and portability approaches in inter-connected clouds: A review. *ACM Computing Surveys 50*, 4 (2017).

[76] Klischewski, R., Tagamoa, A., and Khames, A. Information Integration or Process Integration? How to Achieve Interoperability in Administration. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3183* (2004), 57–65.

[77] Koens, T., and Poll, E. Assessing interoperability solutions for distributed ledgers. *Pervasive and Mobile Computing 59* (10 2019), 101079.

[78] Kolb, J., Abdelbaky, M., Katz, R. H., and Culler, D. E. Core concepts, challenges, and future directions in blockchain: A centralized tutorial. *ACM Computing Surveys 53*, 1 (2 2020).

[79] Kwon, J., and Buchman, E. Cosmos Whitepaper. Tech. rep., Cosmos Foundation, 2016. Available online: https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md, last accessed on 2022-01-10.

[80] LACChain Consortium. LACChain, 2021. Available online: https://www.lacchain.net/home, last accessed on 2021-08-10.

[81] L'Amrani, H., Berroukech, B. E., El Bouzekri El Idrissi, Y., and Ajhoun, R. Toward interoperability approach between federated systems. In *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications* (2017).

[82] Lan, R., Upadhyaya, G., Tse, S., and Zamani, M. Horizon: A Gas-Efficient, Trustless Bridge for Cross-Chain Transactions. Available online: http://arxiv.org/abs/2101.06000, last accessed on 2022-01-10.

[83] Lerner, S. RSK Whitepaper. Tech. rep., RSK, 2015. Available online: https://docs.rsk.co/RSK_White_Paper-Overview.pdf, last accessed on 2022-01-10.

[84] Li, W., and Ping, L. Trust Model to Enhance Security and Interoperability of Cloud Environment. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5931 LNCS* (2009), 69–79.

[85] Lipton, A. Cryptocurrencies change everything. *Quantitative Finance 21*, 8 (2021), 1257–1262.

[86] Litan, A., and Leow, A. Hype Cycle for Blockchain, 2021. Available online: https://www.gartner.com/en/documents/4003463/hype-cycle-for-blockchain-2021, last accessed on 2021-07-22.

[87] Litwin, W., Mark, L., and Roussopoulos, N. Interoperability of multiple autonomous databases. *ACM Computing Surveys (CSUR) 22*, 3 (9 1990), 267–293.

[88] Lohachab, A., Garg, S., Kang, B., Bilal, M., Lee, J., Chen, S., and Xu, X. Towards Interconnected Blockchains: A Comprehensive Review of the Role of Interoperability among Disparate Blockchains. *ACM Computing Surveys (CSUR) 54*, 7 (7 2021), 1–39.

[89] Longo, F., Nicoletti, L., Padovano, A., D'Atri, G., and Forte, M. Blockchain-enabled supply chain: An experimental study. *Computers & Industrial Engineering 136* (10 2019), 57–69.

[90] Lu, J., Yang, B., Liang, Z., Zhang, Y., Demmon, S., Swartz, E., and Lu, L. Wanchain: Building Super Financial Markets for the New Digital Economy, 2017.

[91] Mihaiu, I., Belchior, R., Scuri, S., and Nunes, N. A Framework to Evaluate Blockchain Interoperability Solutions, 2021. Available online: https://www.techrxiv.org/articles/preprint/A_Framework_to_Evaluate_Blockchain_Interoperability_Solutions/17093039, last accessed on 2022-01-10.

[92] Mikhalev, I., Burchardi, K., Struchkov, I., Song, B., and Gross, J. Central Bank Digital Currency (CBDC) Tracker, 2021. Available online: https://cbdctracker.org/cbdc-tracker-whitepaper.pdf, last accessed on 2022-01-10.

[93] Montgomery, H., Borne-Pons, H., Hamilton, J., Bowman, M., Somogyvari, P., Fujimoto, S., Takeuchi, T., Kuhrt, T., and Belchior, R. Hyperledger Cactus Whitepaper. Tech. rep., Hyperledger Foundation, 2020. Available online: https://github.com/hyperledger/cactus/blob/master/docs/whitepaper/whitepaper.md, last accessed on 2022-01-10.

[94] Moon, T., Fewell, S., and Reynolds, H. The What, Why, When and How of Interoperability. *Defence & Security Analysis 24*, 1 (2008), 5–17.

[95] Mühlberger, R., Bachhofner, S., Castelló Ferrer, E., Di Ciccio, C., Weber, I., Wöhrer, M., and Zdun, U. Foundational Oracle Patterns: Connecting Blockchain to the Off-Chain World. In *Lecture Notes in Business Information Processing* (9 2020), vol. 393 LNBIP, Springer Science and Business Media Deutschland GmbH, pp. 35–51.

[96] MYDA. MYDA - Whitepaper, 2021. Available online: https://www.mydacoin.com/whitepaper, last accessed on 2022-01-10.

[97] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. Available online: https://bitcoin.org/en/bitcoin-paper, last accessed on 2022-01-10.

[98] Network, W. Wrapped Bitcoin (WBTC) an ERC20 token backed 1:1 with Bitcoin, 2021. Available online: https://wbtc.network/, last accessed on 2022-01-10.

[99] Pancake Swap. Home | PancakeSwap - $11.911, 2021. Available online: https://pancakeswap.finance, last accessed on 2022-01-10.

[100] Pang, Y. A New Consensus Protocol for Blockchain Interoperability Architecture. *IEEE Access 8* (2020), 153719–153730.

[101] Pasdar, A., Dong, Z., and Lee, Y. C. Blockchain Oracle Design Patterns. *arxiv 2106.09349* (2021).

[102] Pawczuk, L., Gogh, M., and Hewett, N. Inclusive Deployment of Blockchain for Supply Chains: Part 6 – A Framework for Blockchain InteroperabilityPart 6-A Framework for Blockchain Interoperability In Collaboration with Deloitte. Tech. rep., World Economic Forum, 2020.

[103] Pawczuk, L., Massey, R., and Holdowsky, J. Deloitte's 2019 Global Blockchain Survey. Available online: https://www2.deloitte.com/content/dam/insights/us/articles/2019-global-blockchain-survey/DI_2019-global-blockchain-survey.pdf, last accessed on 2022-01-10.

[104] Pillai, B., Biswas, K., and Muthukkumarasamy, V. Cross-chain interoperability among blockchain-based systems using transactions. *Knowledge Engineering Review 35* (2020), 1–18.

[105] Polkadot. Glossary · Polkadot Wiki. Available online: https://wiki.polkadot.network/docs/glossary, last accessed on 2022-01-10.

[106] Polkadot. Polkadot Bridges - Connecting the Polkadot Ecosystem with External Networks. Available online: https://polkadot.network/blog/polkadot-bridges-connecting-the-polkadot-ecosystem-with-external-networks, last accessed on 2022-01-10.

[107] Polkadot. Cross-chain Message Passing (XCMP) · Polkadot Wiki, 2019. Available online: https://wiki.polkadot.network/docs/en/learn-crosschain, last accessed on 2022-01-10.

[108] Polkadot. paritytech/polkadot: Polkadot Node Implementation, 2021. Available online: https://github.com/paritytech/polkadot, last accessed on 2022-01-10.

[109] Polygon. Polygon | Ethereum's Internet of Blockchains, 2021. Available online: https://polygon.technology, last accessed on 2022-01-10.

[110] Putz, B., Dietz, M., Empl, P., and Pernul, G. EtherTwin: Blockchain-based Secure Digital Twin Information Management. *Information Processing & Management 58*, 1 (1 2021), 102425.

[111] Qasse, I. A., Talib, M. A., and Nasir, Q. Inter blockchain communication: A survey. In *ArabWIC 6th Annual International Conference Research Track* (2019).

[112] Qi, M., Wang, Z., Liu, D., Xiang, Y., Huang, B., and Zhou, F. ACCTP: Cross Chain Transaction Platform for High-Value Assets. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (9 2020), vol. 12404 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 154–168.

[113] Quant. Overledger Network for Enterprise, 2021. Available online: https://www.quant.network/overledger-network-for-enterprise, last accessed on 2022-01-10.

[114] Riley, L. Universal DLT interoperability is now a practical reality, 2021. Available online: https://www.hyperledger.org/blog/2021/05/10/universal-dlt-interoperability-is-now-a-practical-reality#.YKzdCEdGRmw.twitter, last accessed on 2022-01-10.

[115] Robinson, P. Survey of crosschain communications protocols. *Computer Networks 200* (12 2021), 108488.

[116] Robinson, P., Hyland-Wood, D., Saltini, R., Johnson, S., and Brainard, J. Atomic Crosschain Transactions for Ethereum Private Sidechains. Tech. rep., 2019. Available online: https://arxiv.org/pdf/1904.12079.pdf, last accessed on 2022-01-10.

[117] Robinson, P., and Ramesh, R. General Purpose Atomic Crosschain Transactions. *arXiv* (11 2020).

[118] Scheid, E. J., Hegnauer, T., Rodrigues, B., and Stiller, B. Bifröst: a Modular Blockchain Interoperability API. In *IEEE 44th Conference on Local Computer Networks* (2 2019), Institute of Electrical and Electronics Engineers (IEEE), pp. 332–339.

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

[120] Siris, V. A., Nikander, P., Voulgaris, S., Fotiou, N., Lagutin, D., and Polyzos, G. C. Interledger Approaches. *IEEE Access 7* (2019), 89948–89966.

[121] Thabet, M., Boufaida, M., and Kordon, F. An approach for developing an interoperability mechanism between cloud providers. *International Journal of Space-Based and Situated Computing 4*, 2 (2014), 88.

[122] TO Group. *ArchiMate® 3.0 Specification.* Van Haren Publishing, 2016.

[123] Tolk, A., and Muguira, J. A. The Levels of Conceptual Interoperability Model. In *Simulation Interoperability Workshop* (2003).

[124] Verdian, G., Tasca, P., Paterson, C., and Mondelli, G. Quant Overledger Whitepaper v0.1. Tech. rep., Quant, 2018. Available online: http://objects-us-west-1.dream.io/files.quant.network/Quant_Overledger_Whitepaper_v0.1.pdf, last accessed on 2022-01-10.

[125] Viho, C., Barbin, S., and Tanguy, L. Towards a Formal Framework for Interoperability Testing. *Formal Techniques for Networked and Distributed Systems* (2 2001), 53–68.

[126] Vo, H. T., Kundu, A., and Mohania, M. Research directions in blockchain data management and analytics. *Advances in Database Technology - EDBT 2018-March* (2018), 445–448.

[127] Weaver. Weaver Interoperability RFCs. Available online: https://github.com/hyperledger-labs/weaver-dlt-interoperability/tree/main/rfcs, last accessed on 2022-01-10.

[128] Wegner, P. Interoperability. *ACM Computing Surveys 28*, 1 (1996).

[129] Wood, G. Polkadot: Vision for a Heterogeneous Multi-Chain Framework. *Whitepaper* (2017), 1–21. Available online: https://github.com/w3f/polkadot-white-paper/raw/master/PolkaDotPaper.pdf, last accessed on 2022-01-10.

[130] World Bank Group. Blockchain Interoperability. Available online: https://www.ft.com/content/1cfb6d46-5d5a-11e9-939a-341f5ada9d40, last accessed on 2021-08-10.

[131] World Economic Forum. Global Standards Mapping Initiative: An overview of blockchain technical standards | World Economic Forum. Available online: https://www.weforum.org/whitepapers/global-standards-mapping-initiative-an-overview-of-blockchain-technical-standards, last accessed on 2022-01-10.

[132] World Economic Forum. Global Agenda Council on the Future of Software & Society Deep Shift Technology Tipping Points and Societal Impact. Available online: https://www3.weforum.org/docs/WEF_GAC15_Technological_Tipping_Points_report_2015.pdf, last accessed on 2022-01-10.

[133] World Economic Forum. Bridging the Governance Gap: Interoperability for blockchain and legacy systems. Tech. rep., 2020. Available online: https://www.weforum.org/whitepapers/bridging-the-governance-gap-interoperability-for-blockchain-and-legacy-systems, last accessed on 2022-01-10.

[134] World Economic Forum. Defining Interoperability – Digital Currency Governance Consortium White Paper Series. Tech. rep., November 2021. Available online: https://www.weforum.org/reports/digital-currency-governance-consortium-white-paper-series, last accessed on 2022-01-12.

[135] Wüst, K., and Gervais, A. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)* (2018), pp. 45–54.

[136] Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A. B., and Chen, S. The blockchain as a software connector. *Proceedings - 2016 13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016* (7 2016), 182–191.

[137] Zamyatin, A., Al-Bassam, M., Zindros, D., Kokoris-Kogias, E., Moreno-Sanchez, P., Kiayias, A., and Knottenbelt, W. J. SoK: Communication Across Distributed Ledgers. Tech. rep., 2019.

[138] Zamyatin, A., Harz, D., Lind, J., Panayiotou, P., Gervais, A., and Knottenbelt, W. J. XCLAIM: A Framework for Blockchain Interoperability. In *IEEE Symposium on Security & Privacy* (2019).

[139] Zarko, I. P., Soursos, S., Gojmerac, I., Ostermann, E. G., Insolvibile, G., Plociennik, M., Reichl, P., and Bianchi, G. Towards an IoT framework for semantic and organizational interoperability. *GIoTS 2017 - Global Internet of Things Summit, Proceedings* (8 2017).