Blockchain Interoperability

Rafael André Pestana Belchior

PhD Thesis Proposal

PhD in Information Systems and Computer Engineering

Advisor: Prof. André Ferreira Ferrão Couto e Vasconcelos
Co-advisor: Prof. Miguel Nuno Dias Alves Pupo Correia

September 2021

Abstract

Blockchain is becoming ubiquitous in today's society. Just in the second quarter of 2021, centralized and decentralized exchanges moved a volume of over $600 billion in cryptocurrencies. Enterprises are adopting this technology, including cryptocurrencies, following the opportunity to expand to new businesses. However, they need to connect their existing systems to blockchains securely and reliably. Blockchain interoperability (BI) is emerging as one of the crucial features of blockchain technology, fueled by the need to eliminate data and value silos.

Given this new domain's novelty and potential, we conduct a literature review on BI by collecting 404 documents. From those 404 documents, we systematically analyzed and discussed 102 documents, including peer-reviewed papers and grey literature. Our review identified four main open problems in the BI research area: 1) lack of systematic solution categorization, 2) lack of evaluation frameworks for BI, 3) gap between theory and practice, and 4) lack of supporting tools for BI. These problems make it challenging for academics and the industry to achieve interoperability among blockchains and centralized systems seamlessly.

Based on the identified problems, the main goal of this thesis is to provide a detailed and extensive approach to blockchain interoperability theory, including classification of solutions, creation of conceptual models, and the design and implementation of blockchain interoperability solutions, supporting tools, and use cases.

In this document, we present the work done so far to address this goal. We propose HERMES, a fault-tolerant middleware that connects blockchain networks and is based on the Open Digital Asset Protocol (ODAP). HERMES is crash fault-tolerant by allying a new protocol, ODAP-2PC, with a log storage API that can leverage blockchain to secure logs, providing transparency, auditability, availability, and non-repudiation. After that, we propose SSIBAC, self-sovereign identity access control, to address identity portability.

Finally, we present the work plan for the rest of this doctoral thesis.

Keywords: blockchain, interoperability, gateways, digital asset, migration, decentralized protocol

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blockchain technology is maturing at a fast pace. The development of real-world applications shows real interest from both industry and academia [3, 4]. For instance, applications have been developed in the areas of public administration [5, 6], access control [7, 8], and others [9].Figure 1.1 depicts the number of search results per year for "BI" that Google Scholar returned. In 2015, only two documents were related to BI. In 2016, 2017, 2018, 2019, and 2020, the results were 8, 15, 64, 130, and 207, respectively, showing a steep increase regarding interest in this research area.

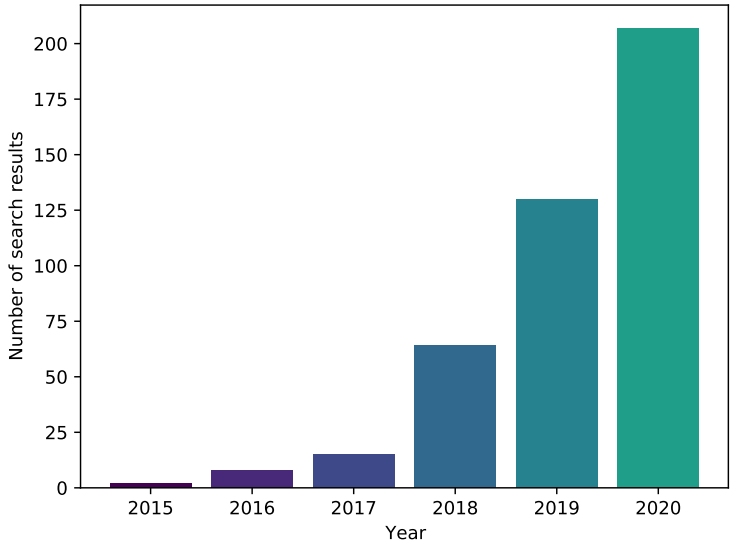

Figure 1.1: Research trends on BI.

Serving multiple use cases and stakeholders requires various blockchain features and capabilities [10]. The need for adaptability is a motivating factor for creating different blockchains, leading to a heterogeneous ecosystem [11]. Choosing new blockchains allows researchers and developers to implement new use case scenarios and keep up with recent endeavors. However, each blockchain has its security risks, as the technology is

still maturing, the user base is limited (e.g., in comparison to the web or databases), and there are uncovered bugs and security flaws [12]. Therefore, developers and researchers have to choose between novelty and stability, leading to a vast diversity of choices [13]. This diversity leads to *fragmentation*: there are many *immature* blockchain solutions (e.g., without extensive testing). Until recently, blockchains did not consider the need for interoperability, as each one focused on resolving specific challenges, leading to *data and value silos* [14].

Moreover, what if the blockchain in which a particular service is running becomes obsolete, vulnerable, or is shutdown? If the user requirements or circumstances change over time, a different blockchain might be more appropriate for a specific use case [15]. What if the service to serve is so crucial that it requires seamless dependability? Furthermore, if we want to reproduce our use case to another blockchain, how can we increase *portability*?

In 1996, Wegner stated that "interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform" [16]. In that context, Wegner established a bridge between the concept of interoperability and existing standards. As those authors were influenced by the standards existing at that time, authors nowadays are influenced by the Internet architecture and concepts, in what concerns BI [17, 18]. Thus, reflecting on the Internet's architecture seems like a good starting point to understand how blockchains can interoperate. Thus, it is essential to solve the *BI* challenge, i.e., to provide interoperability between blockchains in order to explore synergies between different solutions, scale the existing ones, and create new use cases (see Section 2.1.2). For example, a user should be able to transfer their assets from a blockchain to another or build *cross-blockchain decentralized applications*.

While information systems evolve, so do the meaning and scope of interoperability. According to the National Interoperability Framework Observatory (NIFO), endorsed by the European Commission, there are several interoperability layers [19]: *technical interoperability*, *semantic interoperability*, *organizational interoperability*, *legal interoperability*, *integrated public service governance*, and *interoperability governance*. For instance, technical interoperability regards the technical mechanisms that enable integration among blockchains, while semantic interoperability concerns whether the application-specific semantics can be conserved across blockchains. Despite interoperability having an extensive scope, so far most work has been focused on *technical interoperability*, leaving *semantic interoperability* for future work.

Interoperability does not only conflate flexibility and application portability. It also has the potential to solve some of the biggest blockchain research challenges. In particular, interoperability promotes blockchain *scalability*, as it provides a way to offload transactions to other blockchains, e.g., via sharding [20, 21], it can promote privacy (by allowing the end-user to use different blockchain for data objects with different privacy

requirements), and creates new business opportunities.

A recent survey [22] classified BI solutions into three categories: public connectors, hybrid connectors, and blockchain of blockchains. Public Connectors are industry solutions that provide interoperability across public blockchains, focusing on asset transfers. Such solutions are implemented via *sidechains*, *relays*, *hash lock time contracts*, and *notary schemes*. Blockchain of blockchains enables creating customized blockchains that can interoperate by providing reusable data, network, consensus, and contract layers. Hybrid connectors are interoperability solutions connecting public to private blockchains. These solutions are preferred by enterprises as they can integrate blockchains with their legacy systems. Hybrid connectors include *trusted relays*, *blockchain agnostic protocols*, and *blockchain migrators*. Each of these categories responds to a particular set of use cases [22].

Contrarily to public connectors, blockchain of blockchains and hybrid connectors are still in their inception. The Hybrid Connector solution category is the most underexplored, as enterprises are slowly adopting blockchain. Furthermore, the lack of methods to provide data and identity portability and to visualize cross-chain transactions brings resistance to adoption.

1.1 Applications of Blockchain Interoperability

In the second quarter of 2021, decentralized exchanges recorded a volume of $343 billion [23]. Along with Coinbase's total trading volume of $335 billion, the trends towards using blockchain for finance are increasing.

Payment networks, CBDCs and DeFi applications are already being leveraged by multiple players, such as (centralized and decentralized) hedge funds [24, 10]. El Salvador adopted Bitcoin as a legal tender in June 2021. Several dozen projects on central bank digital currencies [1] are gaining traction with the increasing of digitalization of assets and their transfers. Adoption seems inevitable as the world's financial ecosystems evolve [25]. Blockchain provides financial use cases such as cryptocurrency and enables a shared, decentralized, immutable, and transparent record of value. Research suggests that the market for blockchain-based applications will be even bigger, with many organizations stating that blockchain is a critical priority [26, 27], due to, for example, cost reduction. A recent report from Garter predicts that "by 2023, 35% of enterprise blockchain applications will integrate with decentralized applications and services" [28].

Thus, blockchain is slowly but steadily becoming an infrastructure for global value exchange and distributed computation [29]. Amara's Law implies that technologies first undergo underestimation and later mature and unlock their full potential [30]. Blockchains have been created as standalone networks. Moreover, the need to securely and seam-

[1] https://cbdctracker.org/

lessly connect them is still an open problem [31, 32, 33]. Connecting those blockchains (i.e., achieving integration [34]) have a practical utility and importance [10, 31]. It allows communication between systems to exchange data and assets (fungible and non-fungible), leading to a higher heterogeneity of solutions in the market, with more distribution. No blockchain will become a single point of failure. Some of the use cases that require interoperability are digital identity, supply chain, healthcare, and central bank digital currencies (CBDCs) [24, 31]. We believe that blockchain will be adopted not when integration is achieved, but rather *interoperability*. Interoperability is stronger than integration - it allows a system to use the capabilities of other systems in a unified approach [34].

Thus, many applications of blockchain interoperability are still unknown. Network effects and the increasing mass adoption will unlock unexpected paths. However, there are already some use cases for BI. The first important Internet of Blockchains use case is asset transfers, where users can transfer assets from one blockchain to another. While some approaches implement this use case in an ad-hoc way, the emergence of central bank digital currencies (CBDCs) [35], requires further efforts and standardization [36]. A CBDC is a digital version of a sovereign currency of a nation. A CBDC is issued by central banks, where each unit represents a claim on the value held by such a central bank. Many blockchains features are appealing to implement CBDCs, particularly the offered immutability, transparency, and trust distribution. Some central banks are already experimenting with blockchain, including the Monetary Authority of Singapore and the Bank of Canada [37]. As each CBDC can be implemented with a blockchain, and each central bank might choose a different technology, interoperability is achieved using an IoB or even a Blockchain of Blockchains (BoB) approach.

Another major use case is interoperability across supply chains [37, 38]. A supply chain is a chain of value transfer between parties, from the raw product (physical or intellectual) to its finalized version. Managing a supply chain is complex because it includes many non-trusting stakeholders (e.g., enterprises, regulators). As many markets are open and fluid, enterprises do not take the time to build trust - and instead rely on a paper trail that logs the state of an object in the supply chain. This paper trail is needed for auditability and typically can be tampered with, leading to blockchain's suitability to address these problems [10]. A key challenge of blockchain-based supply chains is to interoperate with other distributed ledger technology (DLT) systems. Interoperability granted each supply chain participant (e.g., supplier, manufacturer, retailer) can participate at several supply chains (and thus several blockchains) using a single endpoint, simplifying the interaction process while reducing costs. Other use cases comprise connecting Hyperledger Fabric and Ethereum with Singapore Exchange and Monetary Authority of Singapore via node integration and Evrthng, a product connecting multiple chains via API to digitize products [10].

Finally, identity and data portability can be provided by a multi ledger approach. Iden-

tity paradigms like self-sovereign identity [8] can increase identity portability by providing users control of their identities. Typically, this is achieved by rooting user credentials in a blockchain. Hence, if blockchains can communicate with blockchains' identity providers, one can use the same identity in different blockchains. Data portability complies with blockchains, allowing blockchain users to use their data outside of a blockchain without requiring significant effort.

1.2 Research Scope

This thesis aims to advance the state of art on BI. The starting point for this thesis was asking the question *what is the state of the art on BI?* - which then we investigated in our survey article [22]. We found interesting open research problems:

1. there is a *lack of clarity* in this space, in terms of a general model for BI, solution categorization, and potential of the research area - this leads to adopters having difficulties choosing the right solution.

2. there are *no evaluation frameworks* for BI solutions - this leads adopters to be unable to compare solutions.

3. there is a significant gap between hybrid connectors and other interoperability solutions (*hybrid connectors are underdeveloped* and will be needed) - this hampers enterprise adoption of blockchain.

4. there are *few tools to support BI*, such as abstract data models that represent arbitrary blockchain data, identity portability solutions, and visualization and analysis of cross-chain transactions - also hampering the adoption of blockchain by enterprises.

In this thesis, we explore and contribute to the state of the art of each of these unsolved problems. Our end goal is:

Goal

Create frameworks, models, and tools to analyze, design, and realize BI, respectively.

To achieve this goal, and based on the four identified problems, we identify three key objectives, divided into sub-objectives. For each sub-objective, we explain its relevance and complexity. Each of the sub-objectives addressed in this thesis origins several research questions. By studying these research questions, we hope to cover both the theory and the practice of the emerging research area of BI, providing a solid basis for newcomers. In more detail:

Objective 1 *Understanding the BI research area*

As of early 2020, there was no academic work that systematically reviewed BI literature, covering in-depth the various dimensions we are interested in, namely the necessary background for BI, the terminology used in academia and industry, the architectures, the standards, the classification frameworks, the use cases, the obstacles and the challenges, and the future research directions. We expect to achieve this goal by focusing on the following sub-objectives:

Objective 1.1 *Study available technical requirements for BI*

In this objective, we aim to study the technical requirements for BI. The underlying research questions addressing this objective are i) *What are the minimum set of functionalities a blockchain needs to offer to be able to interoperate with others?* and ii) *What are the minimum set of functionalities a BI solution needs to offer to assure correctness and liveness?*

Objective 1.2 *Explore the current BI solutions available, their strong points, limitations, and future directions.*

In this objective, we aim to study the available solutions through a systematic literature review. The underlying research question is: *How to systematically categorize a BI solution, in terms of its technical characteristics, trust assumptions, and capabilities?*

Objective 1.3 *Create a decision model for BI solutions*

The previous sub-objectives allow a practitioner to understand how solutions classified in terms of performance, advantages, and disadvantages. However, blockchain architects and developers needing to connect blockchains need to 1) understand to which degree their solution is already interoperable with others; 2) choose a particular solution in terms of their functional and non-functional requirements; and 3) test the interoperability solution. To address this problem, we expect to create a decision model for choosing an interoperability solution. The underlying research questions are i) *What are the most common patterns for developing BI solutions?*, ii) *How to assess the interoperability degree of blockchain-based applications and platforms?*, and iii) *How to choose a BI solution?*

Objective 2 *Key Interoperability Enablers*

Blockchain-based applications and blockchain platforms might need a BI solution. If so, the solution needs several building blocks [22]. For example, identity portability is valuable across ledgers because it allows for managing identity across chains, which is a difficult challenge to solve. Likewise, data portability assumes a particular importance because it helps the blockchain applications at the semantic layer by providing semantics

to the data being transferred or migrated. Finally, the systematic analysis of general-purpose cross-chain state is an unsolved (and, as far as we know, an unstudied) problem. For example, it would be helpful to automatically derive cross-chain logic rules among two interoperating blockchains and track several metrics (e.g., performance, end-to–to-end latency, energetic consumption). The visualization of cross-chain transactions could help analytics infer implicit business rules, allowing for a reason for cross-chain logic. Analysis can identify bottlenecks, paving the way to improve performance and cutting costs.

Objective 2.1 *Data Portability*

Data portability is subsumed by the blockchain view concept [39]. A blockchain view is a snapshot of a blockchain from a specific stakeholder. Creating, managing, merging, and safely manipulating blockchain views is still an unsolved problem. The underlying research questions are i) *How to represent data and information across heterogeneous blockchains*, and ii) *How to consolidate different blockchain views?*

Objective 2.2 *Identity Portability*

Identity portability is a key result of the self-sovereign identity (SSI) [8]. We aim to study identity portability from access control (authentication and authorization), since access control will be needed in hybrid connectors. The underlying research questions are i) *Can SSI provide identity portability for Hybrid Connectors?*, and ii) *How can SSI facilitate authorization?*

Objective 2.3 *Analysis Tools*

A virtual shared ledger can be built on top of existing ones [40, 41], allowing it to operate in multiple ledgers. Frameworks creating virtual shared ledgers are referred to as Blockchain of Blockchains, and they can create multiple decentralized applications. These applications can interact with several blockchains using a global state operating in the new trust boundary (i.e., the virtual shared ledger). However, there are no ways of analyzing and visualizing operations on the virtual shared ledger, including but not limited to finding cross-chain rules and visualizing relevant metrics. The underlying research questions are i) *How to visualize cross-chain rules?*, and ii) *What are the relevant metrics to assess cross-chain transactions?*

Objective 3 *Enterprise BI solutions*

Much work has been done in public connectors - interoperability solutions connecting public blockchains. However, as blockchain mature and incorporate enterprise business processes (which seems inevitable), hybrid connectors are needed. There are few hybrid

connectors proper for enterprise use, where privacy, security, interoperability, regulation, standardization, and modularity are essential. We contribute to this research goal with centralized, semi-centralized, and decentralized hybrid connectors.

A conclusion that can generalize to most research areas of computer science is that specific solutions come with trade-offs. With BI, the same happens. A trade-off between scalability, security, and decentralization exists not only on blockchains but also on the new trust boundary connecting them. Two types of hybrid connectors exist, with different trade-offs: centralized, and decentralized.

Summary

The three main objectives compose to deliver our end goal, derived from the main BI open problems. Objective 1 addresses the exploration and comprehension of the state of the art and the development of a general framework to build and evaluate BI solutions. With this knowledge, we can design our interoperability solutions, namely hybrid connectors. However, before that, we need enablers or supporting technologies. In particular, we need the means to migrate data, reuse identity, and analyze cross-chain state (Objective 2). Such foundations allow for a seamless implementation process of enterprise BI solutions, concluding with Objective 3.

1.3 Document Structure

This document is organized into six chapters. For space purposes, we only present our most relevant contributions to date. Each chapter points to the relevant publications.

- **Chapter 2** presents concepts relevant to BI (i.e., the background). After that, we present our framework to classify blockchain interoperability solutions, the *Blockchain Interoperability Framework*. We also present the necessary background on access control and logging for the following chapters for further detail.
- **Chapter 3** presents one of our technical results, SSIBAC, Self-Sovereign Identity Based Access Control, an access control model for cross-organization identity management.
- **Chapter 4** presents another technical result, HERMES, a fault-tolerant middleware that connects blockchain networks and is based on the Open Digital Asset Protocol (ODAP).
- **Chapter 5** presents the plan for the rest of the thesis project.
- **Chapter 6** concludes the thesis research.

Chapter 2

Research Context

In this chapter, we present distributed system and security concepts relevant to blockchain interoperability (i.e., the background). After that, we present our framework to classify blockchain interoperabiliy solutions. This chapter addresses Objective 1 and is supported by publications [22, 8, 42].

2.1 A Primer on Blockchain and Interoperability

The term *blockchain* has at least two different meanings: a type of system and a type of data structure. In this document, we use the term blockchain to denominate a class of distributed systems. A blockchain maintains a shared state, specifically a replicated data structure that we denominate *distributed ledger*. This ledger is maintained by a set of machines with computational and storage resources, called nodes (or peers or participants). Nodes are not trusted individually to maintain the distributed ledger; they are trusted as a group due to their number and diversity [43]. A blockchain can also be considered a *deterministic state machine* that provides a certain service, given existing incentives that the network can reward. The first blockchain was part of the Bitcoin system and provided as service transactions of a cryptocurrency, a digital currency, also designated Bitcoin [2]. The service provided by Bitcoin is the execution of transactions of bitcoins.

Most blockchains are programmable, i.e., their state machine is extensible with user programs. These programs are often designated *smart contracts* [44, 45] and their execution is caused by calls also designated *transactions*. Smart contracts are executed in a virtual machine, e.g., in the Ethereum Virtual Machine (EVM) in Ethereum and other blockchains that adopted the EVM for compatibility (that we designate *EVM-based blockchains*). Smart contracts are often used to implement *tokens*, i.e., blockchain-based abstractions that can be owned and represent currency, resources, assets, access, equity, identity, collectibles, etc. [46]. There are several standard token formats, e.g., ERC-20 and ERC-721 [22]. These tokens are fungible and non-fungible assets, respectively. A fungible asset is interchangeable with another asset of the same type. Conversely, a non-

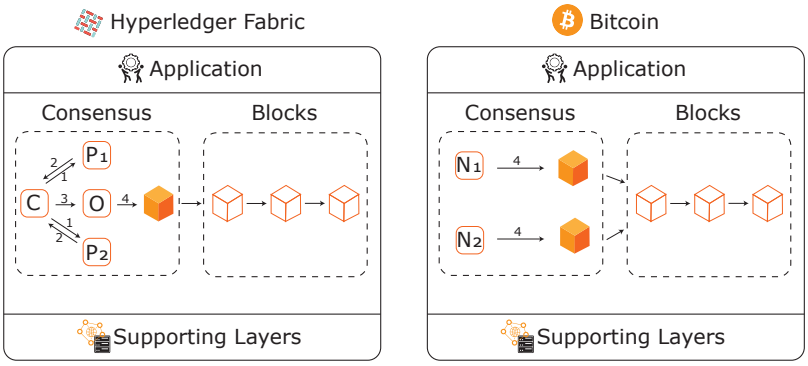

Figure 2.1: Two blockchains: Hyperledger Fabric [1], and Bitcoin [2].

fungible asset is an asset that is unique and has specific properties.

In many blockchains, transactions are aggregated in *blocks*, linked by the previous block's cryptographic hash. Hence those data structures are also called blockchains - often viewed as deterministic state machines.

Blockchain systems ought to be resilient to faults (e.g., *crash fault-tolerant* or *Byzantine fault-tolerant*), as there may be crashes or malicious nodes on the network [47]. They run a consensus algorithm to create agreement on a global ledger state in the presence of Byzantine faults. Consensus algorithms are important because they define the behavior of blockchain nodes and their interaction [47, 48], and the security assumptions of each blockchain. They, therefore, affect how blockchain peers communicate and operate with each other: in Bitcoin's Proof-of-Work (PoW), peers have to compute a cryptographic challenge to validate transactions, competing with each other. Another blockchain, Tendermint, uses a Byzantine fault-tolerant state machine replication (BFT) algorithm [49], supporting up to a third less one of faulty participants. In Hyperledger Fabric, a widely-used private blockchain platform, a consensus algorithm allows higher transaction throughput than PoW by allowing a subset of nodes to execute and endorse transactions (called endorser peers) and by typically using a weaker consensus (only crash fault-tolerant). The variety of blockchain infrastructures makes it challenging to categorize blockchains, and their interoperability solutions, as there are no *de facto* blockchain interoperability or blockchain architecture standards.

Apart from differences in the consensus, blockchains can be deemed *public* (also called permissionless) or *private* (also called permissioned). Permissionless blockchains do not require authentication for participants to access the ledger. *Bitcoin* [2] and *Ethereum* [45, 50] are examples of such blockchains. Permissioned blockchains are blockchains in which users are authenticated and can be held accountable according to a governance model suitable for enterprise and governmental needs. Hyperledger Fabric [1], Corda [51], Quorum [52], Tendermint [49], and Multichain [53] are examples of permissioned blockchains.

Figure 2.1 depicts two blockchains: Hyperledger Fabric, a permissioned blockchain; and Bitcoin, a permissionless blockchain. The supporting layers (e.g., networking, storage, encryption) [54] provide a basis for the consensus engine, which orders transactions and appends them to the chain of blocks. In Hyperledger Fabric, the consensus is modular, based on endorsement policies. In Fabric, a client (C) sends a transaction proposal to the peer nodes (P), and obtains a signed transaction, called an endorsement (steps 1 and 2). An orderer validates the endorsements and builds a block with valid transactions, appending it to the ledger (steps 3 and 4). In Bitcoin, the consensus is based on the notion of Proof-of-Work (PoW), a cryptographic puzzle that mining nodes need to solve in order to build a valid block. This corresponds roughly to Fabric's steps 1-3. After a node finds a solution to PoW, it then can propose a block of transactions to be appended to the ledger (step 4).

Blockchain trust is based on the incentive models that guide the behavior of the nodes. For instance, in Bitcoin, nodes have the incentive to produce blocks of transactions and support the network because they are rewarded Bitcoins. Conversely, nodes do not have the incentive to disrespect the protocol, as attacks are expensive and nodes can get punished [55]. In Hyperledger Fabric, where nodes are identified, they have the business incentive to follow the protocol because parties cooperate towards a common goal, and misbehavior can be punished according to the law or applicable governance model. Decentralization, different goals, and incentives support the trust on the blockchain – parties can share the ledger without relying on a trusted, centralized party.

The ability to distribute trust on a global state fostered the appearance of *decentralized applications* (*dApps*) [46]. A dApp is a computer program running on a decentralized peer-to-peer network. For example, Steemit[1] is a social blogging dApp that rewards content-creators with cryptocurrency. Thus, dApps are based on smart contracts running on a blockchain, but they also have other components that should equally be decentralized.

2.1.1 Cross-Blockchain Communication

Cross-blockchain communication involves two blockchains: a *source blockchain*, and a *target blockchain*. The source blockchain is the blockchain in which the transaction is initiated to be executed on a target blockchain. While general-purpose interoperability comes down to a blockchain exposing its internal state to other, cross-chain asset transfers rely on an atomic three-phase procedure: 1) locking (or extinguishing) of an asset on a source blockchain; 2) blockchain transfer commitment, and 3) creation of a representation of the asset on a target blockchain [56, 57, 20]. This procedure, later explained in detail, relies on a *cross-chain communication protocol* (CCCP).

[1] https://steemit.com/

A CCCP defines the process by which a pair of blockchains interact to synchronize cross-chain transactions correctly. Hence, a CCCP allows *homogeneous* blockchains to communicate. For instance, sidechains typically use a CCCP (e.g., Zendoo allows communication between Bitcoin-like blockchains systems [58]). Conversely, a *cross-blockchain communication protocol* (CBCP) defines the process by which a pair of blockchains interact to synchronize cross-blockchain transactions correctly. CBCPs allow *heterogeneous* blockchains to communicate (e.g., the Interledger Protocol allows any blockchains that implement the protocol to exchange "money packets" [59]). The differentiation between CCCPs and CBCPs is important because CCCPs typically can leverage the interoperating blockchains' constructs and functionality (e.g., utilize smart contracts to implement a relay [60]), whereas CBCPs normally require blockchains to be adapted. However, CBCPs may leverage specific functionalities of both blockchains [61]. Cross-blockchain, or cross-chain communication, is a requirement for blockchain interoperability. This section provides a few theoretical results regarding cross-blockchain communication, and thus also blockchain interoperability.

Zamyatin et al. [62] prove that "there exists no asynchronous CCC [cross-chain communication] protocol tolerant against misbehaving nodes". The authors use a reduction to the fair exchange problem [63] to prove that correct cross-chain communication is as hard as the fair exchange problem. As a consequence of the presented theorem, the authors state that "there exists no CCC protocol tolerant against misbehaving nodes without a trusted third party". A trusted third party can be centralized or decentralized. Centralized trusted parties are, for example, trusted validators [38]. A decentralized trusted party can be another blockchain, in which their participants agree on the global ledger state via a consensus algorithm. However, the trusted party has to ensure that most participants are honest, guaranteeing the correctness of the process is guaranteed. Cross-chain protocols, therefore "use the consensus of the distributed ledgers as an abstraction for a trusted third party." [62]. Borkowski et al. [64] derive the "lemma of rooted blockchains" that states that a source blockchain cannot verify the existence of data on a target blockchain with practical effort. In particular, the source blockchain would need to be able to mimic consensus from the target blockchain, and it would have to store a (potentially large) subset of the target blockchain's block history. On a recent endeavor, Lafourcade and Lombard-Platet [65] formalize the blockchain interoperability problem, arguing that fully decentralized blockchain interoperability is not possible. More specifically, there is no protocol assuming a full-client that can realize its interoperability functions, such as asset transfer, without a third party's aid. However, a blockchain with two ledgers offers the possibility of interoperability (there is, in fact, the possibility of moving assets from one ledger to the other). This study applies mainly to public blockchains.

The results above are relevant because they lead to an important consideration: *cross-blockchain transactions are not feasible in practice without the participation of a trusted*

third party. In other words, although trust assumptions vary greatly from permissionless to permissioned networks, cross-blockchain transactions, as well as cross-chain transactions, require a trusted third party to assure the correctness of the underlying protocol.

2.1.2 Blockchain Interoperability Definitions

In this section, we define additional technical terms for an understanding of this study.

Vernadat defines interoperability among enterprise systems as [66]: "a measure of the ability to perform interoperation between [...] entities (software, processes, systems, business units...). The challenge relies on facilitating communication, cooperation, and coordination among these processes and units". Abebe et al. propose a general communication protocol as an alternative approach to the "point-to-point" blockchain interoperability approach [14]. Interoperability is defined as "the semantic dependence between distinct ledgers to transfer or exchange data or value, with assurances of validity". Pillai and Biswas refer that "cross-communication is not intended to make direct state changes to another blockchain system. Instead, cross-communication should trigger some set of functionalities on the other system that is expected to operate within its own network" [67].

A technical report from the National Institute of Standards and Technology (NIST) defines blockchain interoperability as "a composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain are reachable, verifiable, and *referable* by another possibly foreign transaction in a semantically compatible manner" [68]. Hardjono et al. define blockchain survivability as "the completion (confirmation) of an application-level transaction [composed of subtransactions] independent of blockchain systems involved in achieving the completion of the transaction."[17] The concept of transactions and subtransactions relates to "*best effort delivery*", that applications must comply to, by ensuring that transactions and their *subtransactions* are completed (i.e., committed) within a certain time frame.

Regarding types of blockchain interoperability, Besançon et al. highlight three [69]: interoperability between different blockchains, interoperability between dApps using the same blockchain, and interoperability blockchain and other technologies (such as integration with enterprise systems). While different definitions tackle different dimensions of interoperability, there is room for improvement. We define several terms that encompass the whole scope of technical interoperability to later provide a holistic definition of technical interoperability (see Figure 2.2). To recall the definition presented in Section 2.1.1, a source blockchain is a blockchain that issues transactions against a target blockchain. A *source node* is a node from the source blockchain, and a target node belongs to the target blockchain. When several participants elect a source node and a target node, we achieve decentralization in the context of interoperability [70].

A CC-Tx, where "CC" stands for *cross-chain*, and "Tx" for transaction, is a transaction between different chains, which belong to the same blockchain system (homogeneous blockchains), for example, between EVM-based blockchains. We use the CC-Tx, *inter-chain transaction*, and *inter-blockchain transaction* terms interchangeably. A CB-Tx is a transaction between different blockchains (heterogeneous blockchains), for example, between Hyperledger Fabric and Bitcoin. Note that the terms CC-Tx and CB-Tx are used as synonyms in the industry, as currently, most solutions connect homogeneous blockchains. A CC-dApp is a dApp that leverages cross-blockchain transactions to implement its business logic. We use the terms CC-dApp and *cross-blockchain decentralized application* (CB-dApp) interchangeably. Other terms with the same meaning in the literature are inter-chain decentralized application and inter-blockchain decentralized application.

A IoB is a system "where homogeneous and heterogeneous decentralized networks communicate to facilitate cross-chain transactions of value" [18]. We use this definition of IoB throughout this paper.

The term BoB is not used consistently [41, 71]. Verdian et al. use it to describe the structure that aggregates blocks from different blockchains into "meta blocks", organized through a consensus mechanism using *posets* (partially ordered sets) and total order theory [72], thus producing a blockchain of blockchains. A poset consists of a set of elements and their binary relationships that are ordered according to a specific set of rules [72].

Influenced by those authors, we define a BoB *as a system in which a consensus protocol organizes blocks that contain a set of transactions belonging to CC-dApps, spread across multiple blockchains. Such a system should provide accountability for the parties issuing transactions on the various blockchains and providing a holistic, updated view of each underlying blockchain*. Note that BoB solutions belong to the category with the same name. Therefore, the notion of IoB directly refers to the connection relationships among blockchains, whereas the term BoB refers to an architecture made possible by IoB. BoB approaches are concerned with the validation and management of cross-blockchain transactions.

Figure 2.2 shows the relationship between the different concepts concerning blockchain interoperability. A CC-dApp realizes the blockchain of blockchains approach. This approach can provide the semantic level interoperability (i.e., concerned at transmitting the meaning of the data, which corresponds to the value level interoperability) required by organizations, mappable by the applicational layer. However, it relies on the existence of an IoB – a network of blockchains. For an IoB to exist, technical interoperability (or mechanical interoperability) is required. In the context of a CC-dApp, cross-chain transactions are ordered by a *cross-chain dApp protocol*. Such protocols should assure transaction atomicity and resolve possible conflicts in transactions spawning across homogeneous and heterogeneous blockchains.

From the several definitions we encountered during our research, we envision *blockchain*

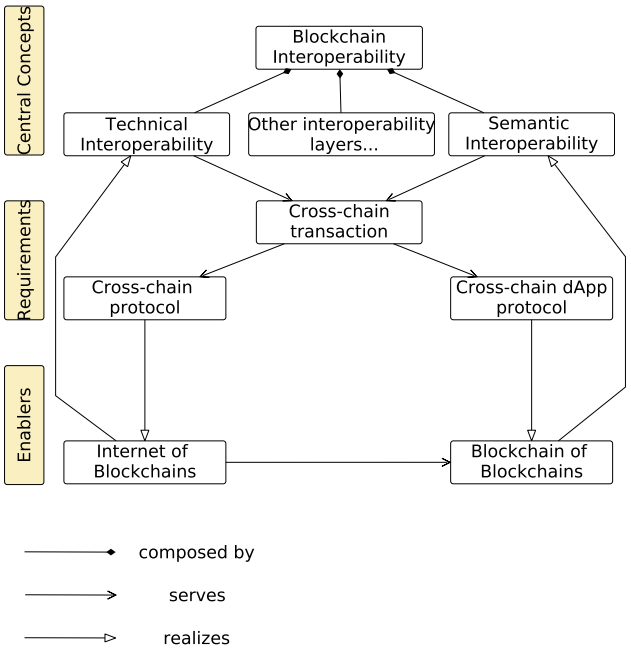

Figure 2.2: Concept map, illustrating the relationship between different concepts related to blockchain interoperability

interoperability as: *the ability of a source blockchain to change the state of a target blockchain (or vice-versa), enabled by cross-chain or cross-blockchain transactions, spanning across a composition of homogeneous and heterogeneous blockchain systems, the IoB.* IoB transactions are delivered via a cross-blockchain communication protocol, thereby granting technical interoperability, enabling CC-dApps. CC-dApps provide semantic interoperability via the BoB. The BoB approach is realized by a cross-blockchain dApp protocol, which provides consensus over a set of cross-chain transactions, thus enabling cross-chain dApps.

2.1.3 Blockchain Interoperability Framework

This section presents the Blockchain Interoperability Framework (BIF), a framework classifying solutions collected through our systematic literature review (please consult [22] for details). To drive criteria for assessing the categories (and specific solutions) of blockchain interoperability, we analyzed the solution space using the six "W" questions: Who, What, Where, When, Why, and How. The "Why" was determined irrelevant to our analysis because its purpose is constant – connecting different chains (CC-Txs), different blockchains (CB-Txs), or even to arbitrary systems (e.g., enterprise legacy systems). This is instead addressed by the "where" question.

Deriving Evaluation Criteria

The "what" refers to the *assets* exchanged. An interoperability solution can handle different data objects or assets. Hence it is important to know which data representations a solution supports [16]. Assets can be treated as data (arbitrary payloads), as fungible assets, or non-fungible assets [73, 74, 38]. Arbitrary data is often represented via a key-value pair, being the preferred representation of some blockchains [1, 43]. The key-value is also useful to represent the contents of account-based blockchains [75, 76, 52]. Payment tokens are fungible tokens [67]. Utility tokens include tokens used to access a service or application, such as non-fungible tokens (e.g., ERC20 tokens). Finally, asset tokens represent real-world physical or digital instruments, such as blockchain-based promissory notes, regulated by the Swiss Financial Market Supervisory Authority [37], or bonds [73]. An asset has different maturity levels. In particular, an asset may be standardized, (e.g., ERC tokens[77], standardized schema for utility tokens, ERC1400, a security token [78]) and/or regulated [79, 80, 81]. Regulated digital assets are backed by legal frameworks. We consider all asset tokens to be regulated. We envision utility tokens as standardized and asset tokens as standardized and regulated (i.e., asset tokens are emitted by legal entities).

The "who" question refers to whom controls the CC-Tx process and thus accounts for trust establishment [82, 62]). It can be the end-user (e.g., [38, 83]), a consortium (e.g., [84, 85]), or a trusted third party (e.g., cloud services, centralized notary schemes). Some solutions allow different levels of control.

The "where" refers to what are the source and target ledgers, as well as what is the support of conducting the CC process. Solutions can support public blockchains (P) or non-public blockchains (NP). We use NP to designate private blockchains, other decentralized ledger technology (DLT) systems, and centralized systems (e.g., VISA payment network). The supported systems of each solution matter since communication may happen unidirectionally or bi-directionally [38]. Blockchain oracles apart, it often is not feasible to have a solution based on a blockchain system connected to a centralized system (e.g., providing insurance data). A smart contract may be the one conducting an asset transfer (on-chain channel, with on-chain CC-Tx validation) versus an off-chain settlement, e.g., techniques using commitment schemes [86, 58], or via (semi-)centralized system (off-chain channel). Typically, on-chain channels offer more resiliency, but off-chain channels are more scalable. Combinations between off-chain and on-chain channels also exist (e.g., payment networks [87]). Offline channels depend on different proof generation mechanisms [86, 82, 58].

The "when" refers to the set of processes (e.g., executing CC-Txs) that are defined at *design-time* or *run-time*. *Design-time customization* decisions affect the punctual behavior of a CC-dApp concerning when it is executed. At design-time, a user defines the behavior of the solution *apriori*. If a change is needed, a new instance of the solution needs to be de-

ployed. Conversely, *run-time customization* decisions are flexible, allowing the end-user to adjust the conditions defined by business logic as needed. Solutions in which business logic is changed at run-time are called *flexible approaches*, allowing to adjust business logic and conditions that trigger the execution of a CB-Tx or CC-Tx by a CC-dApp. Most literature reviews focus on design-time approaches and public blockchains, leaving a vast range of recent solutions out of scope. In this survey, we also consider private-private and public-private blockchain interoperability, focusing on flexible approaches.

The "how" regards the realization of cross-chain transactions: how are CC-Txs realized on the underlying DLTs? Often, these transactions can be performed using *cross claims*, i.e., by locking/burning an asset on the source blockchain and unlocking/creating its representation on the target blockchain. Cross-claims require two nodes from different blockchains, where one performs one operation in a source blockchain in exchange for its counterparty performing other operations on a target blockchain - each party logs the operation in case a dispute is needed. Typically, cross-claims operate in semi-trusted environments (e.g., private blockchain, regulated blockchain), and can be operated via a (semi) trusted third party [38, 88, 89]. Escrowed cross-claims are the standard mechanism for asset transfers, operating similarly to cross-claims, but in an untrusted environment, leveraging dispute-resolution mechanisms (e.g., via smart contracts requiring inclusion proofs [86]) or by parties holding custody of assets and collateral, [90, 91, 92]. Inclusion proofs include applying Merkle tree proofs to block header transfer via a coordinating blockchain, block header transfer, or direct signing [93]. Collateralization is the process in which a party performing the transfer of assets provides a certain amount of their assets as a guarantee of following the protocol (e.g., not to steal assets from the end-user). If a party misbehaves (e.g., steals assets), the deposit is given to the victim party. Finally, a mediated CC-Tx includes (an offline) trusted party [38]. In case of a dispute about an asset transfer between a public blockchain and a private blockchain (P-NP) or a public blockchain and an enterprise system (also P-NP), there needs to be a dispute-resolution mechanism. This is due to NP systems' private nature, although several mechanisms exist to prove internal state belonging to private blockchains. Hence, CC-Txs have a trade-off risk-performance: the less centralization there is on the CC-Tx settlement, the worst the performance, but the lesser the risk.

The "how" also relates to the extent to which the implementation of the solution is tested. Solutions might be implemented, tested, and validated (application to a real-world scenario). Testing regards *correctness guarantees*: *behavioral correctness* or *formal correctness*. Behavioral correctness is the ability to guarantee that CC-Txs are issued as intended, without unintended consequences (e.g., asset lock, asset theft). While in practice, behavioral correctness depends on formal correctness, we say a solution has behavioral correctness if it has a suite of test cases [94]. Formal correctness assures that an algorithm is correct with respect to a specification. Formal verification checks the correctness of al-

gorithms against a specification using, for instance, formal methods. Smart contract verification tools allow developers to reduce the probability of creating bugs, thus incurring penalties, as smart contracts are generally difficult to update once deployed [95]. Another point of providing trust to the user is the solution to have an open-source implementation, where the code can be peer-reviewed and corrected if needed.

Evaluation Criteria

We now define the set of criteria we use to characterize the interoperability solutions. Each criterion can be "fulfilled" "partially fulfilled" or "not fulfilled". If a criterion is a yes/no question (e.g., does the solution support asset type "data"?), we do not explicitly refer to the fulfillment conditions as they are evident. Next, we detail the criteria type (first-level), criteria sub-type (second level), and criteria from BIF:

- Asset: this category refers to properties of an asset involved in a CC-Tx.
  - Type: what type of assets does the solution support?
    1. Data: can the solution manipulate arbitrary data?
    2. Payment tokens: can the solution manipulate cryptocurrencies? This criterion is partially fulfilled if the asset is only used as collateral or to reward a service's operational maintenance.
    3. Utility tokens: can the solution manipulate utility tokens? This criterion is partially fulfilled if the asset is used only as collateral or to reward a service's operational maintenance.
    4. Asset tokens: can the solution manipulate utility tokens?
  - Infrastructure: what are the systems involved?
    1. P: This criterion is fully fulfilled if more than two public blockchains are supported. It is partially fulfilled if one or two public blockchains are supported.
    2. NP: This criterion is fully fulfilled if more than two non-public blockchains are supported. It is partially fulfilled if one or two non-public blockchains are supported.
- Trust Establishment: this category refers to how a solution provides trust to the users.
  - Decentralization: who operates the solution instance?
    1. End-user
    2. Consortium
    3. Trusted (third) party

    If multiple criteria are selected, it indicates a solution supports more than one mode of operation.
  - Channel: where are CC-Tx validated?
    1. On-chain: This criteria is partially fulfilled if proofs are created on-chain but validation occurs off-chain.
    2. Off-chain: This criteria is partially fulfilled if proofs are created off-chain but validation occurs on-chain.

- CC-Tx Realization: this category refers to how and where a CC-Tx is settled.
  - Mechanism: how are CC-Txs agreed-upon multiple parties?
    1. Cross-claim
    2. Escrowed cross-claim
    3. Mediated
- Extra-functional: this category refers to the design of the solution itself.
  1. Tests: the approach provides a set of test cases.
  2. Implementation: the approach provides an open-source implementation and is validated in the industry. This criterion is partially fulfilled if the implementation is closed-source.
  3. Validation: the approach is validated in an actual use case scenario.
  4. Run-time: the business logic of the solution can be changed dynamically, as needed. This criterion is considered not fulfilled if logic is settled when the solution is instantiated, i.e., changing logic requires a new instance.

Overview of Blockchain Interoperability Approaches

We conducted a systematic literature review, yielding 80 relevant documents out of the initial 404. By grouping the publications and grey literature, a pattern arises: these works are either about interoperability across public blockchains holding cryptocurrencies, application-specific blockchain generators with interoperability capabilities, or protocols connecting heterogeneous blockchains. We thus classify each study into one of the following categories: *Public Connectors*, *Blockchain of Blockchains*, and *Hybrid Connectors*. Each category is further divided into sub-categories. Table 2.1 summarizes the work conducted.

The first family of blockchain interoperability solutions, public connectors (in green) aim to provide interoperability between cryptocurrency systems, as stated by Vitalik [141]. This category identifies and defines different chain interoperability strategies across public blockchains supporting cryptocurrencies, including sidechain approaches, notary schemes, and hash time hash-locks.

Blockchain of Blockchains are frameworks that *provide reusable data, network, consensus, incentive, and contract layers for the creation of application-specific blockchains (customized blockchains) that interoperate between each other.* We briefly present Polkadot [125] and Cosmos [49], the most widely adopted Blockchain of Blockchains in terms of market capitalization[2].

The *Hybrid Connector* category is composed of interoperability solutions that are not Public Connectors or Blockchain of Blockchains. Directed to both public and private blockchains, Hybrid Connectors attempt at delivering a "blockchain abstraction layer"

[2]USD 22.1B and USD 3.6B respectively, as of February 2021

	Asset			Trust Establishment										
	Type			Infra.		Decentral.			Channel		CC-Realization			
Sub-Category	D	P	U	P	NP	U	C	TTP	OC	OF	CC	ECC	M	**References**
Sidechains & Relays	+	±	-	±	-	-	+	-	+	+	-	+	-	[96, 97]
	+	±	-	±	-	+	+	-	+	-	-	+	-	[61, 60, 98, 99, 100]
	-	+	+	+	-	+	+	-	+	-	-	+	-	[101, 102]
	-	+	+	±	-	+	+	-	+	+	-	+	-	[84, 103, 104, 105, 106]
	+	+	-	±	-	-	+	-	+	-	-	+	-	[107, 108, 58, 109]
	-	+	+	±	-	-	+	-	+	±	-	+	-	[110, 111, 112, 113, 114, 115]
	-	+	-	+	-	+	+	+	-	+	+	-	+	[59, 116]
Notary Scheme	-	+	+	+	-	-	-	+	±	-	-	-	+	See survey [22]
	-	+	+	+	-	+	+	-	+	-	-	+	-	[117, 118, 119]
HLTC	-	+	+	±	-	-	+	-	+	-	-	+	-	[91, 90, 120, 121, 122, 123, 124]
Blockchain of Blockchains	+	+	+	±	-	+	+	-	+	-	-	+	-	[49, 125, 126]
	+	+	+	+	+	-	+	+	+	-	-	+	+	[127, 128, 41]
Trusted Relays	+	-	-	±	±	+	-	-	-	+	-	-	+	[129, 130, 54, 131]
	+	+	+	±	+	+	+	-	+	±	+	-	+	[17, 57, 88, 89, 132, 133, 134]
B. Agnostic Protocols	+	+	+	+	+	+	+	-	-	+	+	+	-	[38, 86, 14, 135]
	+	+	+	±	±	+	+	-	+	-	-	+	-	[71, 136, 74, 93, 137, 138]
Blockchain Migrators	+	-	-	±	-	+	-	-	-	+	N/A	N/A	N/A	[83, 139, 140]
	+	+	+	±	±	+	+	-	+	-	-	+	-	[20]

Table 2.1: Evaluation of blockchain interoperability solutions by subcategory accordingly to the Blockchain Interoperability Framework. N/A means not applicable. Public connectors are in green, Blockchain of blockchains in orange, and Hybrid connectors in red.

[10], capable of exposing a set of uniform operations allowing a dApp to interact with blockchains without the need of using different APIs [130].

Due to space limitations, we defer the explanation of each sub-category to [22].

2.2 Self Sovereign Identity and Access Control

This section introduces concepts regarding self-sovereign identity (decentralized identifiers, verifiable credentials) and access control.

2.2.1 Self-Soverign Identity

To empower the user with control over of his data, while proving dynamic, trustable, and decentralized ACMs, we refer to the concept of *Self-Sovereign Identity* (SSI) [142]. SSI is a good match to the blockchain promise of decentralization [143, 133]. In SSI, the user stores identity data and decides which data to disclose. Unlike existing schemes such as OpenID Connect (OIDC) [144], Shibboleth [145], and Microsoft Passport [146], there is no need to entrust an intermediary identity provider with storing identity data [147]. SSI can alleviate the impact of data breaches and provide the user with flexibility managing the identity: instead of spreading data and information among different service providers,

the user has full control of his personal data and discloses only required information. By using *zero-knowledge proofs* (ZKPs), SSI allows satisfying predicates based on user data without revealing that data [148]. This provides privacy for access control processes, where a user needs to satisfy a certain predicate to access resources. SSI also allows a single identity to be linked to sets of attributes emitted by different organizations. Therefore, it fosters interoperability across administrative domains and applications [149].

Blockchain is a suitable technology to support SSI, as it is decentralized and supports peer-to-peer interaction [143]. Furthermore, it can be used to obtain a reliable infrastructure for decentralized access control, mitigating some of its traditional problems, such as the lack of adaptability to dynamic environments [150]. Although the use of a replicated immutable appendable log could raise concerns regarding the GDPR, SSI allows technical privacy protection, achieving GDPR compliance [143]. In particular, SSI does not compromise GDPR's view on the right of users to rectify and remove data and promote the identification and regulation of data processors. Conversely, the application of SSI to the access control process can also protect users' privacy.

SSI provides a model for *authentication and issuing credentials*, rooted on the following concepts: *decentralized identifiers* (DIDs) [151], and *verifiable credentials* (VCs) [152].

2.2.2 Decentralized Identifiers

The SSI concept allows a user – individual, organization, or "thing" (e.g., a device or a computer program representing a process) – to present its credentials to a third party without intermediaries. This process is enabled by DIDs, a concept defined by the W3C [151]. A DID represents an identity and allows trustable interactions, rooted on a *verifiable registry* (e.g., a blockchain), and public-key cryptography. DIDs are controlled by *DID subjects*. A DID resolves to a DID document with metadata, which also provides the means for authenticating the DID subject. The DID subject can prove the ownership of a DID through a private key associated with a DID's public key. A DID can be defined as a three-part string representing the format `did:<method>:<identifier>`, where `<method>` represents the DID method (the specification for a specific type of DID) that the `<identifier>` uses [151]. To facilitate the management of DIDs, one can leverage *user agents* (or simply agents), i.e., software processes acting on behalf of a DID subject [148].

2.2.3 Verifiable Credentials

A *verifiable credential* (VC) provides a standard way to digitally express credentials in a way that is cryptographically secure, privacy-respecting, and machine-verifiable [152, 148]. An entity called *issuer* generates and signs such credentials with its private key:

this enables a third-party to verify the issuer of a VC (the DID of the issuer is typically associated with the credential). A *verifier* can look up the public key of a given DID, associated with a given credential on a verifiable data registry (e.g., a public blockchain).

For example, the VON ledger[3] is a Hyperledger Indy-based blockchain storing public DID documents, *credential definitions* (representing the schema of a VC, i.e., the attributes the VC should hold) and revocation registries (repositories containing information about revoked credentials). *Credential schemas* allow a verifier to check the *claims* against a vocabulary of admissible claims. A claim is an assertion on a subject. For example, an issuer defines a set of possible attributes in a schema that may later be issued in VCs associated with that schema. A VC, therefore, consists of claims made about a subject by an issuer. The subject and issuer are represented by unique identifiers, which we assume to be DIDs for our purposes. A VC is trusted as long as its issuer is trusted.

At verification time, the holder of a verifiable credential (often the subject itself) creates a *verifiable presentation* (VP), which contains metadata and proofs for a subset of the contained claims. The VP creation process might be required by a verifier, through a *verifiable presentation request* (VPR). The VP is sent to a verifier, that confirms the VC held by the subject satisfies a specific predicate. The VP can be issued using ZKPs, "containing derived data instead of directly embedded verifiable credentials" [152]. For simplicity, we deem that the result of a generated VP can be true or false, if the predicate is satisfied or not, respectively.

2.2.4 Access Control

Access control systems provide selective access to a set of resources, under a specific set of conditions. Common ACMs include *Role-Based Access Control* (RBAC) [153] and *Attribute-Based Access Control* (ABAC) [154], besides many others, e.g., the classical Access Control Matrix, Access Control Lists, Capabilities, Mandatory Access Control, and Discretionary Access Control [155].

In ABAC, a commonly used ACM, access rights are granted based on attributes, i.e., the attributes the subject holds and the attributes expressing the environmental context. According to the XACML specification [156], which is suitable to implement an ABAC system, several components are cooperating in the access control process. The *subject* (that we also call user) is the entity that requires access to a resource. A *client* is a device that requests access to a resource on behalf of a subject. A *Policy Enforcement Point* (PEP) intercepts access requests from a user, redirecting them to the *Policy Decision Point* (PDP), and enforcing its AC decision. The PDP is the component that computes the result of an access control request (ALLOW or DENY), using an access control policy and information stored on the *Policy Information Point* (PIP). The PIP contains information about the subject's attributes. The *Policy Retrieval Point* (PRP) stores and retrieves

[3] `https://vonx.io/`

access control policies, which are managed by the *Policy Administration Point* (PAP). Although the literature separates the attribute storage (PIP) from the access control policy storage (PRP), we refer to them as the same entity, for brevity. Moreover, *accountability* is achieved by tracking the access control requests issued by the subject, and the corresponding access control decision calculated by the PDP. This process allows the system to establish a history of access to resources. Moreover, centralized *access control* systems face several challenges and risks [150, 157]: cumbersome policy management, lack of flexibility of setup and configuration, ineffective policy enforcement, risk of privacy leakage, and availability (single point of failure). These translate into issues of *authentication*, *authorization*, and *accountability* (AAA).

2.2.5 Centralized and Federated Identity

Enterprise identity systems typically focus on roles or attributes associated with each user, enabling the execution of their duties. In enterprise identity and access management (IAM) the friction caused by centralized systems is most apparent in federation scenarios, where external users have to be granted access to internal systems. An example is *Eduroam*, a federation of educational organizations that provide internet access to each others users [158]. Traditionally, this is achieved by requesting data through identity federation systems [159]. However, identity federation systems are not interoperable among different standards and bridges are required to interact across federations [160].

2.3 Atomic Commit Protocols and Logging

An atomic commit protocol (ACP) is a protocol that guarantees a set of operations being applied as a single operation. An atomic transaction is indivisible and irreducible: either all operations occur, or none does. ACPs consider two roles: a *Coordinator* that manages the execution of the protocol, and *Participants* that manage the resources that must be kept consistent. ACPs assume stable storage with a write-ahead log (a history of operations are persisted before actions are executed). Example of ACPs are the two-phase commit protocol, 2PC, the three-phase commit protocol, 3PC, and non-blocking atomic commit protocols [161].

2PC achieves atomicity even in case of temporary system failure, accounting for a wide adoption both in the academia and in the industry. It has two phases: the voting phase and the commit phase. In the voting phase, the coordinator prepares all participants to take place in a distributed transaction by inspecting each participant's local status. Each participant executes eventual local transactions required to complete the distributed transaction. If those are successful, participants send a *YES* response to the coordinator, and the protocol continues. Else, if the *NO* response is sent, it means that the participant chose to abort; this happens when there are problems at the local partition. Next, in

the commit phase, when the coordinator obtains *YES* from all participants, a *COMMIT* message is sent to the participants that voted *YES*. This message triggers the execution of local transactions that implement the distributed transaction. Otherwise, the coordinator sends an *ABORT* message, triggering a rollback on each local partition.

A log $\mathcal{L}$ is a list of log entries $\{l_1, l_2, ..., l_n\}$ such that entries have a total order, given by the time of its creation. A log is considered *shared* when a set of nodes can read and write from the log. On the other hand, a log is *private* (or *local*) when only one node can read and write it. Logs are associated to a process p running *operations* on a certain node. To manipulate the log, we define a set of *log primitives*, that translate *log entry requests* from a process p into log entries. The log primitives are *writeLogEntry* (writes a log entry), *getLogLength* (obtains the number of log entries), and *getLogEntry(i)* (retrieves a log entry l_i). A log entry request typically comes from a single event in a given protocol.

A *log storage API* provides access to the primitives. Log entry requests have the format `<phase, step, operation, nodes>`, where the field *operation* corresponds to an arbitrary command, and the field *nodes* to the parties involved in the process p. We define four operations types to provide context to the protocol being executed. Operation type *init-* states the intention of a node to execute a particular operation, and operation *exec-* expresses that the node is executing the operation. The operation type *done-* states when a node successfully executed a step of the protocol, while *ack-* refers to when a node acknowledges a message received from another. Conversely, we use the type *fail-* to refer to when an agent fails to execute a specific step. The field *nodes* contains a tuple with a node A issuing a command, or a node A commanding a node B the execution of a command c, if the form is A or $A \rightarrow B$ (c may be omitted), respectively.

Chapter 3

Enabling Identity Portability with SSIBAC

Ineffective data management practices pose serious issues to individuals and companies, e.g., risk of identity theft and online exposure. Self-sovereign identity (SSI) is a new identity management approach that ensures users have full control of their personal data. SSI provides a model for *authentication and issuing credentials* for cross-organization identity management. To implement SSIBAC, we leverage three emerging technologies: blockchain, *decentralized identifiers* (DIDs) [151], and *verifiable credentials* (VCs) [152].

SSIBAC leverages conventional access control models and blockchain technology to provide decentralized authentication, followed by centralized authorization. The access control process does not require storing user sensitive data. A prototype was implemented and evaluated, processing 55,000 access control requests per second with a latency of 3 seconds.

We show how DIDs and VCs can be integrated with attribute-based access control in a federated setting, minimizing data disclosure and data redundancy. For transparency and accountability regarding access requests, VCs can be used with blockchain-based ACMs.

We go beyond existing work on attribute-based access control by ensuring user privacy. While privacy has been a concern for ABAC models [159], existing ABAC models still store all identity data with a single identity provider. With SSIBAC, selective disclosure of attributes and range proofs for numerical values ensure that data is only disclosed on a need-to-know basis.

This chapter addresses Objective 2 and is supported by the following publication [8]. In short, this chapter contributions are:

- a novel SSI-based ACM called SSIBAC, with a focus on data privacy and sovereignty;
- an implementation of SSIBAC, relying on an attribute-based model;
- an evaluation of the implementation, providing insights on the bottlenecks of the solution and future research directions; the prototype processed 55,000 access control

requests per second with a latency of 3 seconds.

Chapter Outline: This chapter is organized as follows: We formally define the SSIBAC model in Section 3.1, followed by an instance of such model using ABAC in Section 3.2. We report and discuss the evaluation results in Section 3.3. Finally, in Section 3.4 we summarize and conclude the chapter.

3.1 The SSIBAC Access Control Model

This section presents the SSIBAC access control model. The SSIBAC model is an evolution of classical ACMs that integrates the concept of SSI and mechanisms that implement it with blockchain. The major idea of SSIBAC is to map VCs (encoded into VPRs, and their responses, VPs) to access control policies, stored at the PRP, that are parsed by an underlying ACM, in order to achieve context-based privilege, and thus data privacy and sovereignty.

SSIBAC abstracts previous models and can be instantiated using one of those models, e.g., RBAC or ABAC. This means that a particular instantiation of SSIBAC reuses concepts and mechanisms of the underlying model. SSIBAC regulates the access of subjects to resources by evaluating access control rules against *permission validators*. Permission validators allow mapping VPs to attributes, roles, or other abstractions of data. For instance, if SSIBAC is instantiated with RBAC, the permission validator is the role, whereas if SSIBAC is instantiated with ABAC, the permission validator is the set of subject and contextual attributes. A user is uniquely identified by a DID (although a user can hold multiple DIDs), and has a set of VCs, issued by *issuers*. An issuer is a trusted entity that issues VCs.

A permission validator, along with an access control request, allows the PDP to calculate an access control decision. We consider a function ψ that maps VCs to permission validators, depending on the input. For example, $\psi_{(i,ATTRIBUTE)}$ maps all the user verifiable credentials from user$_i$ to attributes that can be used by an ABAC system. Conversely, $\psi_{(i,ROLE)}$ maps the VCs from $user_i$ to roles, which can be evaluated by an RBAC system. In practise, the initialization of ψ depends on the underlying access control system to be used, and its mapping is trivial. We, therefore, establish the bridge between DIDs, VCs, and the permission validators of ACMs, by saying "this VC corresponds to an attribute/role defined in a specific AC policy". This function can be considered the component that facilitates interoperability among ACMs, similarly to meta-access control models [162].

We define a function χ that maps access control policies to VPs. This function bridges the access control policies used by conventional ACMs with peer-to-peer interactions supported by a trusted data verifier. Function χ can be defined in 1) an ad-hoc way, 2) automated by parsing the schema fields and creating a VP containing the same fields,

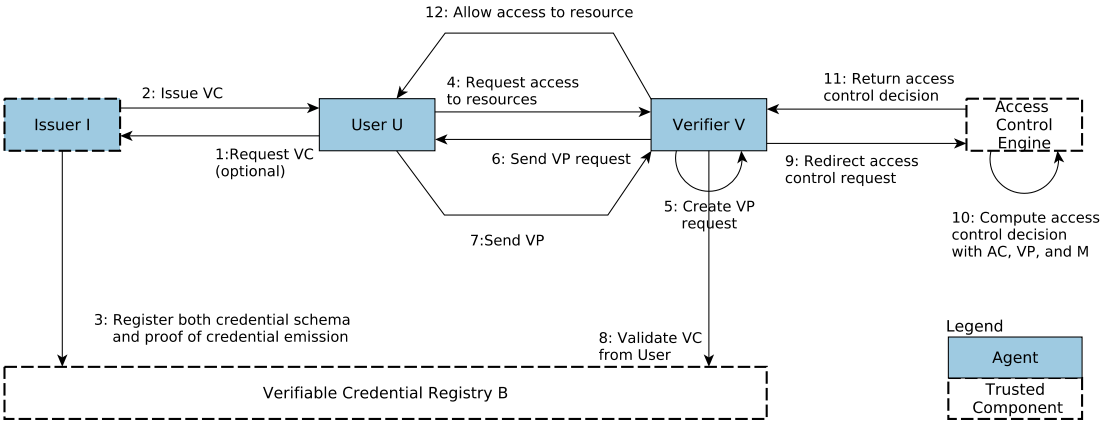

Figure 3.1: Access control flow enforced by the SSIBAC model

plus a condition defined on the access control policy. Verifiable presentations are then tied to an access control request, requested by verifiers. Verifiers can also be providers of resources (i.e., the entity processing the access control request is the same that holds and delivers the resource).

The infrastructure supporting the issuing of DIDs, VCs, and VPs is a verifiable data registry (in our specific case, a verifiable credential registry). This data registry can be decentralized, e.g., a blockchain. For instance, a blockchain can record the schema of a verifiable credential, along with its issuer: this allows peer-to-peer validation of VCs without resorting to the issuer.

We define our model formally as follows:

SSIBAC components:

- a set of users $\mathcal{U} = \{u_1, u_2, ...\}$. Each $user_i$ is identified by a DID and holds a public/private key pair (K_p^i, K_s^i) associated to that DID and a set of VCs $\mathcal{L}_i = \{l_1^i, l_2^i, ...\}$;
- a set of resources $\mathcal{R} = \{r_1, r_2, ...\}$;
- a set of issuers $\mathcal{I} = \{i_1, i_2, ...\}$ that issue VCs for users;
- a set of verifiers $\mathcal{V} = \{v_1, v_2, ...\}$ who request VPs and mediate the access control flow. Typically, they are also resource providers;
- a set of permission validators $\mathcal{P} = \{p_1, p_2, ...\}$;
- a set of injective functions $\psi = \{\psi_1, \psi_2, ...\}$, such that $\psi_i : \mathcal{L}_i \rightarrow \mathcal{P}_k$, i.e., function ψ_i maps the VCs from $user_i$ to permission validator P_k;
- an injective function $\chi : \mathcal{AC} \rightarrow \mathcal{VP}_R$, mapping access control policies to VPRs.

Besides the core components, SSIBAC has several input parameters, which can be set before instantiation or computed at run-time.

SSIBAC parameters:

- a set of supporting ACMs $\mathcal{M}$;
- a set of access control policies $\mathcal{AC}$, representing the rules of a particular business context;
- a set of VPs $\mathcal{VP}$, translated from access control policies;
- a Verifiable Data Registry $\mathcal{B}$, the trust anchor for the peer-to-peer interactions (allows checking the validity of DIDs, VCs, and VPs).

Figure 3.1 illustrates the access control flow enforced by our model. A user is issued verifiable credentials (steps 1 and 2), which are rooted in a verifiable credential registry (3). The user then requests access to a set of resources (4). The verifier creates a VPR from the access control policy underlying the requested resource. This access control policy may be collected from a trusted PRP (5) and sends it to the user (6). The verifier assumes that the user owns the necessary attributes on the verifiable credentials to be able to respond to the challenge. After the challenge is sent in the form of a VP (7) and validated (8), the verifier gives as input the result of the validation process to an access control engine (or PDP) (9, 10). The result of the decision may be influenced by extra factors, e..g. the context of the request. If the decision from the access control engine is ALLOW, access to the resource is provided (11, 12).

3.2 SSIBAC instantiated with ABAC

In this section, we describe *SSIBAC instantiated with the ABAC model*. We chose ABAC because it is much adopted and provides fine-grained and flexible access control [163].

3.2.1 System Description and Assumptions

To integrate decentralized identity with attribute-based access control, attributes need to be issued to a specific DID. This can be accomplished using VCs [148].

We instantiate the SSIBAC model with $\mathcal{M} = \{ABAC\}$, a user u_1, an issuer i_1, a verifier v_1, one resource r_1, a public blockchain $\mathcal{B}$. The permission validator p_1 is the attribute from ABAC ($\psi_1 : \mathcal{L}_1 \rightarrow \{p_1\}$) . In other words, our access control engine will calculate an access control decision based on ABAC/XACML access control policies, so we need VPs to encode user attributes. Let $\mathcal{L}_1$ represent the subset of verifiable credentials held by user u_1, and let $\Lambda_1 = \{\lambda_1, ..., \lambda_i\}$ be a subset of attributes derived from $\mathcal{L}_1$.

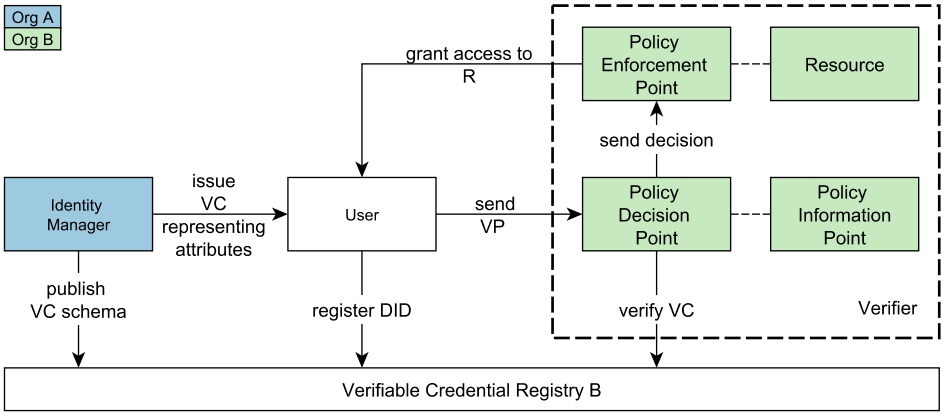

Figure 3.2: SSIBAC in a multi-organizational setting, in light of the XACML standard perspective

Function χ maps a verifier's access control policy AC_1, containing the rules to access $r1$, to a VPR, by parsing the schema fields from the VC(s), as well as the access control policy, and the necessary conditions for an ALLOW decision. The VPR is issued by v_1, while the corresponding VP, VP_1 generated by u_1. Access to a certain resource is granted given that v_1 returns an ALLOW decision, under the condition that the result of the VPR, VP_1, is true. In other words, the AC_1 encoded by VPR, and evaluated by VP_1, is satisfied.

The access control decision could be comprised of a more complex policy, e.g., the result of VP_1 and a set of contextual conditions, such as the day of the week. For this, the PIP could be hybrid: sensitive data is owned by the subject, whereas general information used to identify him is also saved on a local database. User attributes are mapped to the verifiable credentials emitted to a specific DID, i.e., the subject holding ownership of the DID, with a specific schema. The access control policies are mapped to VPRs made on-chain. By doing so, we allow users to keep their information private, as verifiable presentations can handle selective disclosure based on zero-knowledge proofs [148].

Figure 3.2 shows how ABAC components are integrated within SSIBAC. One can observe that the verifier acts as both PEP and PAP. It allows or denies access to resources through access control policies, and administrates such policies. The access control engine contains a PDP that can be embedded in the verifier, or be an external component. We opted for a centralized PDP, although decentralized ones are possible and have been implemented [163].

We remark that a decentralized PIP may be used. Sensitive information for enabling access control decisions can be held by the subject. A combination of a subject-owned PIP and a traditional, local PIP can be useful: sensitive attributes can be kept private by the subject, while other attributes such as the user ID are stored by the organization.

3.2.2 Threat Model and Security Requirements

Regarding SSIBAC's threat model, we assume an honest-but-curious verifier. This means that the verifier performs the access control decision honestly, but may try to learn about the users' attributes. Since verifiable presentations are supported by ZKPs, the verifier will, very likely, obtain incomplete information – selective disclosure is achieved. However, selective disclosure is usually not enough, as organizations can collude to cooperatively infer information about the user. Thus, *unlinkability* is also desired. Our model achieves unlinkability given that a person utilizes a DID for each specific purpose.

The security requirements are threefold:

1. *Selective choice of participants*: only users holding the VCs which map to the permission validators required in an access control policy can access the resources specified on the same access control policy.

2. *Data confidentiality*: the access control engine should perform decisions based on the least information possible. The ZKPs allows a user to disclose as least information as possible.

3. *Accountability and non-repudiation*: issuers are held accountable for the VCs they issue. User credentials are auditable, as the blockchain provides the trust anchor for checking its validity. In other words, a verifier can verify that the presented credentials are valid and come from a trusted party, at its description. We provide a trade-off between privacy and accountability as the interactions between DIDs are peer-to-peer and thus not necessarily recorded; unlinkability is established if a DID interacts does not interact with several parties, disclosing (part) of their VCs.

3.3 Evaluation

In this section, we evaluate an instance of SSIBAC based on the real-world use case scenario from the European Commission (EC) project QualiChain.[1]

3.3.1 Use Case: Decentralised Qualifications

The QualiChain project aims to propose a blockchain-based approach for disrupting the archiving, management, and verification of educational and employment qualifications. In particular, QualiChain will support the storage, sharing, and verification of academic and other qualifications along with several additional services, provided by the platform. To comply with GDPR legislation, and protect its users' privacy and data, a non-intrusive access control mechanism has to be deployed. In particular, QualiChain aims to follow the

[1] https://qualichain-project.eu/

principle of *context-based privilege*, in which only the strictly necessary data to provide a service is requested from the diploma holder.

This project has several stakeholders:

- *certification seekers*, e.g., graduated students. They are referred to as diploma holders upon receiving a verifiable credential for their diploma;
- *certification providers*, e.g., higher education institutes;
- *certification validators*, e.g., potential employers.

Universities issue verifiable credentials for students, which can be used to authenticate on the QualiChain platform. It is desirable to use SSI-based access control in this scenario so that QualiChain does not need to store any personal data: access to services is provided on-demand, based on the verifiable proofs that the student provides.

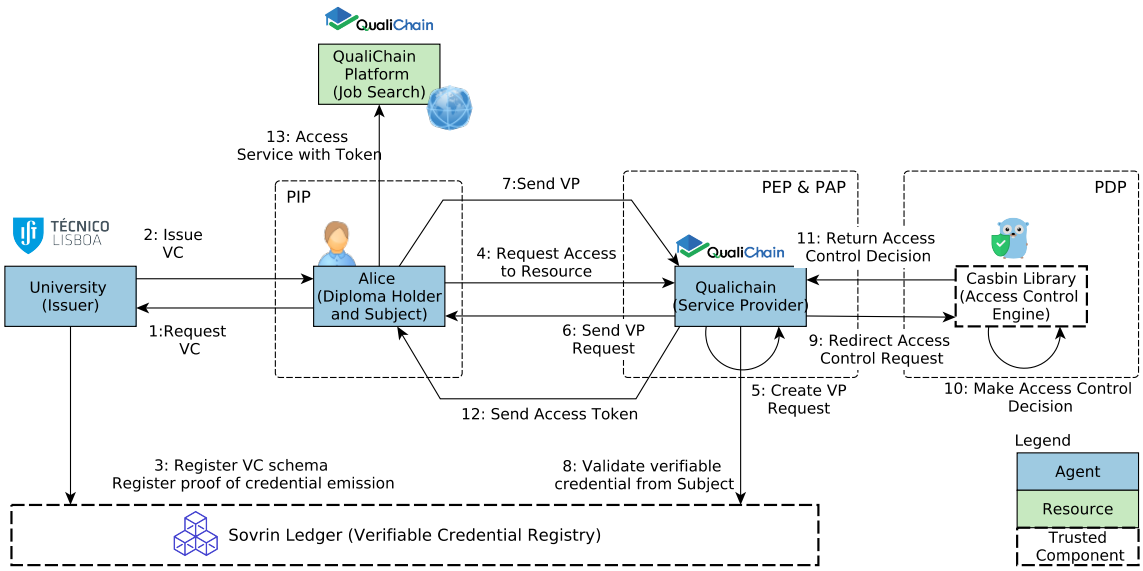

Figure 3.3: SSI-based ACM applied to the QualiChain scenario

SSIBAC can be useful for access control in QualiChain. We focus on granting a diploma holder access to a service provided by QualiChain. Figure 3.3 illustrates this process.

We instantiate our model defined in Section 3.2 with $u_1 = Alice$, $i_1 = IST$, $v_1 = QualiChain$, and $r_1 = JobOffers$, a service provided by the QualiChain Platform. In this use case, a recent graduate, Alice, requires a university diploma in the form of a VC from a higher education institution, IST (step 1). The university issues a VC to Alice, and publishes the corresponding proof on a decentralized ledger, in our case the Sovrin blockchain based on Hyperledger Indy (steps 2 and 3). Listing 1 depicts the *credentialSubject* schema issued for the student. Alice now becomes a diploma holder.

```
"firstName": "Alice",
"lastName": "Anderson",
"age": 25,
"id": "1234",
"timestamp": 1590092610,
"degree": {
    "university": "IST",
    "type": "BachelorDegree",
    "name": "Bachelor of Science", "EQF": "6",
    "course": "Computer Science",
    "grade": "4",
    "gradeScale": "0-4",
    "skills": "[]",
    "degreeId": "80970"
},
"metadata": [...],
"proof": [...]
```

Listing 1: High-level example of a verifiable credential, issued for Alice, $\mathcal{L}_{Alice}$. Λ_{Alice} = $\{$*firstName, LastName, ... , degreeId*$\}$, $p1$ = attribute (VC mapped to attributes by χ), χ parses the VC fields such that it outputs a VP containing a challenge invoking a subset of Λ_{Alice}

Upon accessing the platform, which may require Alice a VP certifying she owns a non-revoked VC issued by IST, Alice can have access to several services. We consider the job offer service, enabling PhD diploma holders to find research positions, available for people with a European Qualification Framework[2] level higher than 6 (MSc and PhD graduates).

Alice would like to access the service that allows searching for job offers on research positions (step 4). In order to provide her access to that service (a resource), QualiChain creates a VPR and sends it to Alice (steps 5 and 6). The VP encodes the access control policy for accessing the researcher position service: if the "EQF" field of a diploma holder is higher than 6, access to the service is provided. Alice constructs and provides the corresponding ZKP (step 7), and the QualiChain agent verifies that such VP is true (step 8). After validating that the proof comes from Alice, it redirects the result to the access control engine (step 9), which later returns the processing outcome (steps 10 and 11). The access control policy states that if the result from the VP is true, then access is granted. However, additional checks could be performed (e.g., the access control engine could verify if Alice had already accepted another job offer). If the verifiable presentation is valid, it means that it satisfies the encoded access control policy sent in step 6. As Alice's EQF is not higher than 6, Alice cannot access the desired service (steps 12 and 13 do not take place).

[2]https://www.cedefop.europa.eu/en/events-and-projects/projects/european-qualifications-framework-eqf

3.3.2 Implementation

We now describe the implementation of the SSIBAC prototype and the experimental setting.

As our Verifiable Credential Registry $\mathcal{B}$, we chose Hyperledger Indy [164]. Hyperledger Indy is a state-of-the-art public blockchain that provides "tools, libraries, and reusable components for providing digital identities". The Hyperledger Aries project [165] was leveraged to create agents (representations of users) that manage their wallets and perform operations on the distributed ledger. Aries serves as the infrastructure for "blockchain-rooted, peer-to-peer interactions". In other words, we mediate communication between agents and the supporting blockchain through Aries. We based our implementation on the demo provided by Hyperledger Aries.[3]

We ran our experiments on the GreenLight Dev Ledger[4] provided by the VON blockchain test net. We leveraged Google Cloud Platform as our infrastructure. A c2-standard-8 (8 vCPUs, 32 GB memory) virtual machine running Ubuntu 20.04 was used. After the appropriate setup, we deployed three Docker containers, each representing an agent: Alice, IST, and QualiChain. A fourth agent was deployed to aid the evaluation process, by collecting information on the performance of the process. We executed each experiment 100 times and discarded the first and the last 10 to avoid outliers. In total, we executed 400 experiments.

3.3.3 End-to-End Latency

First, we measure the end-to-end latency of the process. For that, we divide our process into three phases: startup, connect, and access control.

The *Startup* phase comprises the time to set up the necessary infrastructure for conducting access control based on decentralized identity. In particular, this phase includes the time the system needs to register the agents' DIDs into the blockchain, the time to initialize the four agents (Alice, IST, QualiChain, and Performance), and the time for the IST agent to publish the schema of a university degree.

The *Connect* phase connects the agents, exchanges verifiable credentials, and prepares the environment for the access control phase. Alice connects to IST, IST issues a variable number of verifiable credentials to Alice, and after that posts a corresponding proof on the blockchain. Next, Alice connects to the QualiChain agent.

In the *Access Control* phase, Alice requests a resource from QualiChain. QualiChain requests a verifiable presentation (output of χ) that contains the necessary permission validators (attributes Λ_1, according to ψ) in order to conduct the access control process (conducted by ABAC). Alice constructs the proof and sends it to QualiChain. QualiChain

[3] https://github.com/hyperledger/aries-cloudagent-python
[4] http://dev.greenlight.bcovrin.vonx.io/

then handles the proof, confirms its validity, and conducts the access control process.

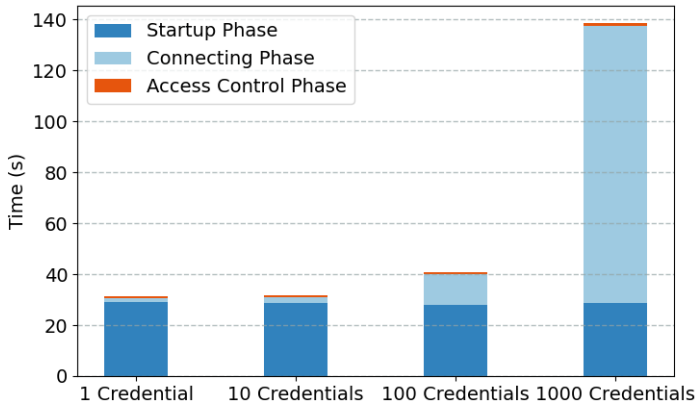

Figure 3.4: Latency depending on the number of emitted credentials

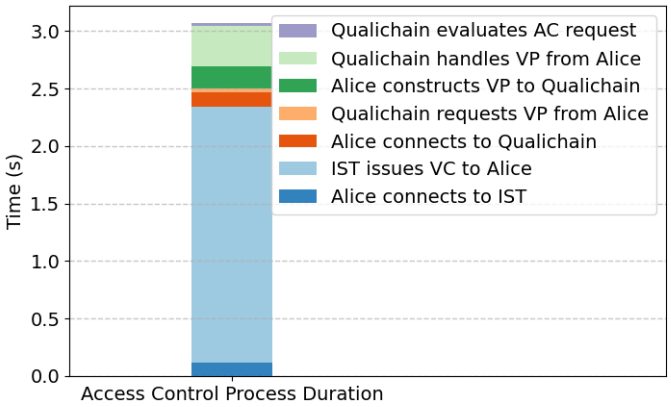

Figure 3.5: Duration of the various steps in the *Connecting* and *access control* phases (startup phase omitted), with 10 issued credentials

Figure 3.4 depicts the cumulative time necessary to conduct each phase, in seconds, as a function of the number of credentials emitted to Alice. The *Startup* phase takes 29.1, 28.5, 28.0, and 28.7 seconds if the number of verifiable credentials issued was 1, 10, 100, or 1000, respectively. Conversely, the *Connect* phase took 1.3, 2.5, 12.0, and 109.0 seconds. A linear regression for these results yields the function $t(n) = 1.283 + 0.108n$, where $t(n)$ is the *Connect* time and n the number of credentials. The access control phase took 0.9 seconds regardless of the number of credentials. Table 3.1 depicts the latency associated with each phase of the access control process, as a function of the number of credentials issued. One can observe that the average duration of the *Startup* and *Access Control* phases is practically constant. The constant startup phase duration is expected, as no step of this phase depends on the number of issued credentials.

Table 3.1: Evaluation of latency as a function of the number of credentials

# CredentialsProcess Phase	Startup		Connecting		Access Control		Total Time
	μ	σ^2	μ	σ^2	μ	σ^2	
1 Credential	29.14	5.5	1.30	0.03	0.92	1.05	31.36
10 Credentials	28.53	5.24	2.47	0.09	0.92	1.05	31.92
100 Credentials	28.00	4.85	11.97	0.33	0.92	1.05	40.89
1000 Credentials	28.68	4.84	108.93	0.92	0.92	1.05	130.53

The *Startup* phase is a major bottleneck, accounting for most of the duration of the overall process (92.9%, 89.3%, 68.4%, and 21.9% for 1, 10, 100 and 1000 issued credentials respectively). For a system issuing a large number of VCs, the *Connect* phase is the bottleneck. As shown before, the system scales linearly with the number of credentials emitted. As credentials will not be emitted regularly, the *Connect* phase duration can be significantly reduced. Figure 3.5 represents the sub-phase duration for the *Connect* and *Access Control* phases. The overall process takes 3.39 seconds (considering 10 issued credentials), being the credential issuing step responsible for 66% of the connecting phase and access control phases together (2.23 out of 3.39 seconds).

For a single credential, the first access control request can take up to around 31.4 seconds (*Startup*, *Connecting*, and *Access Control* phases) or 2.23 seconds (*Connecting* and *Access Control* phases) to be served. In particular, the *Startup* and *Connecting* phases require setting up infrastructure and peer-to-peer connections specific to our experimental setting, which explains high latencies. In practice, the full process of credential issuance and a subsequent access control request should normally take around 2.23 seconds. In the QualiChain scenario, the attributes do not vary frequently, so we deem this latency suitable.

3.3.4 Throughput

The throughput in terms of access control requests per second is associated with the latency of the previous phases. Credentials are issued during the *Connecting* phase, with about 9.3 credentials issued per second for 1000 credentials (see Table 3.1). This limit is due to sequential processing in our demo application, since all credentials are sequentially created, signed and submitted. The throughput performance is also tied to the hardware in which the experiment is running and the throughput of the Hyperledger Indy consensus algorithm.

The time for evaluating the access control policy is negligible, as we achieve around 55,000 access control evaluations per second considering only the *Access Control Phase*. Considering that the necessary credential for the verifiable presentation has been emitted and belongs to the subject, the time needed for processing each access control decision is 0.9 seconds.

3.3.5 Revocation

Credential revocation is an important concern for access control systems. If a credential is revoked (e.g., the university revokes Alice's diploma), the verification of the presentation by the verifier will fail. Thus, when Alice attempts to access a resource using her revoked VC, access will be denied. This does not incur a performance degradation of our system, as the process is the same with a valid credential, and revoking a credential only costs a transaction.

3.4 Summary

Identity portability will play an important part on interoperability between blockchains, as authentication and authorization processes should utilize the same identities across networks. In this Chapter, we contributing to solve this problem with SSIBAC, the first approach to access control based on decentralized identity. We explore this topic by instantiating our SSIBAC model with attribute-based access control, which is applied to a real-world case, the EU QualiChain project. Our implementation assures that the context-based privilege is achieved, by promoting peer-to-peer interactions, providing the basis for the access control process. Our experimental evaluation shows that each access control request can be served in around 0.9 seconds. Although more time-consuming than traditional centralized access control systems, access control based on self-sovereign identity can alleviate the data privacy problem, which we consider an acceptable trade-off for applications not requiring high throughput.

Chapter 4

Enabling Cross-Jurisdiction Digital Asset Transfer with Hermes

Enabling blockchain-based digital asset exchanges requires BI capabilities. Although significant progress on interoperability has been made, public blockchains, private blockchains, and legacy systems cannot communicate seamlessly yet [22]. Moreover, current solutions are not standardized and do not offer the possibility to seamlessly transfer data and value across legal jurisdictions, hampering enterprise adoption of blockchain. There is a need for building solutions capable of complying with legal frameworks and regulations.

We believe that similar to Internet routing gateways, which enabled interoperability around private networks, and fostered the rise of the Internet, the global network of decentralized ledgers (DLTs) will require blockchain gateways [166, 167]. Gateways permit digital currencies and virtual assets to be transferred seamlessly between these systems. Within the Internet Engineering Task Force (IETF), there is currently ongoing work on an asset transfer protocol that operates between two gateway devices, the *Open Digital Asset Protocol* (ODAP) [166]. ODAP is a cross-chain communication protocol handling multiple digital asset cross-border transactions by leveraging asset profiles (the schema of an asset) and the notion of gateways. Transferring an asset between blockchains via gateways is equivalent to an atomic swap that locks an asset in a blockchain and creates its representation on another. However, how can one guarantee a fair exchange of assets (either all parties receive the assets they requested, or none do) across gateways?

To assure the properties that enable a fair exchange of assets, blockchain gateways must operate reliably and be able to withstand a variety of attacks. Thus, a crash-recovery strategy must be a core design factor of blockchain gateways, where specific recovery protocols can be designed as part of the digital asset transaction protocol between gateways. A recovery protocol, allied to a crash recovery strategy, guarantees that the source and target DLTs are modified consistently, i.e., that assets taken from the source DLT are persisted into the recipient DLT, and no double spend can occur.

To realize this vision, we propose HERMES, a fault-tolerant middleware that connects blockchain networks, enabling the transfer of data and value across legal jurisdictions.

HERMES is based on the Open Digital Asset Protocol (ODAP), an asset transfer protocol. HERMES utilizes a novel mechanism called ODAP-2PC and decentralized logging that can solve disputes regarding asset exchange. We find HERMES to fill an existing gap: the technical infrastructure that can constitute the basis for legislating and regulating cross-chain transfers, enabling the future of finance.

This chapter addresses Objective 3 and is supported by the following publication [8]. In short, this chapter contributions are:

- we present HERMES fault-tolerant middleware, instantiated with the ODAP protocol and ODAP-2PC.
- we provide a comprehensive discussion on HERMES as a solution for BI, focusing on consistency, performance, and decentralization.
- we briefly explore a use case for cross-jurisdiction asset transfers, illustrating how one can leverage HERMES to achieve BI compliant with legal and regulatory frameworks.

Chapter Outline: This chapter is organized as follows: we introduce the gateway concept and HERMES, in Section 4.1. After, in Section 4.2, we present ODAP-2PC. Section 4.3, presents a use case that benefits from HERMES. Section 4.4 presents our discussion on gateways, ODAP, and ODAP-2PC in the light of the presented research questions. Finally, we conclude this chapter in Section 4.5.

4.1 The Architecture of HERMES

This section introduces the gateway concept and HERMES.

4.1.1 Blockchain Gateways

A *gateway* is a DLT system node based on an underlying DLT-based system and functionally capable of performing CC-Tx, including asset transfers [167]. A *primary gateway* is the DLT system node acting as a gateway in a CC-Tx. Primary gateways may be supported by *backup gateways* for fault tolerance. For gateways to be crash fault-tolerant, they keep track of each operation they do in a log (of operations). The log is a sequence of log entries, each entry representing a step of the gateway protocol. A gateway protocol specifies the set of messages and procedures between two gateways for their correct functioning. The gateway protocol considered in this chapter is ODAP [166, 168].

4.1.2 Blockchain Interoperability with HERMES

HERMES is a gateway system that enables DLT interoperability based on gateways. This system has four layers, allowing for end-to-end communication. The gateway protocol

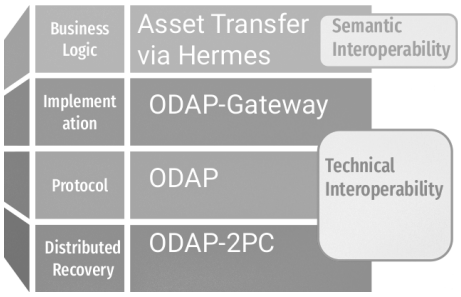

Figure 4.1: Hermes architectural layers

layer implements any standards that a specific gateway implementation needs to comply with (e.g., travel rule [11]). ODAP, a gateway-based CCCP that realizes asset transfers, allows realizing technical interoperability for asset transfers. It is built on top of a distributed recovery protocol, providing reliability in the presence of crashes. On top of the gateway protocol stands a concrete implementation of a gateway. Jointly with the gateway protocol, it provides support for semantic interoperability [22], unlocking the value level. More specifically, in the value level, the business logic is defined for clients using gateways, allowing them to attribute value to the assets exchanged with ODAP. The whole stack provides atomicity, consistency, isolation, and durability of CC-Tx. Figure 4.1 represents HERMES' architectural layers.

Our architecture is flexible and modular, as its components are pluggable. Modularity allows building a system that can be adapted to specific needs. In this chapter, we instantiate HERMES with the ODAP-Gateway, the ODAP CCCP, and its crash fault-tolerant distributed recovery protocol, ODAP-2PC. The whole stack allows a business case, gateway-to-gateway asset transfers, providing the basis for unidirectional asset transfers, expressed in detail in Section 4.3. The HERMES Client allows to implement the business logic, realizing semantic interoperability.

4.2 Hermes

In this section, we present the main building blocks of HERMES: ODAP and HERMES' distributed recovery mechanism, ODAP-2PC.

4.2.1 ODAP and Properties

The ODAP protocol is a gateway-to-gateway unidirectional asset transfer protocol that uses gateways as the systems conducting the transfer [166]. An asset transfer is represented in the form $T : G_1 \overset{a,x}{\rightarrow} G_2$, where a source gateway G_1 transfers x asset units from type a from a source ledger $\mathcal{B}_S$ to a recipient ledger $\mathcal{B}_R$, via a gateway G_2.

The source gateway issues a transfer such that x asset units will be unavailable at the source DLT and become available at the target DLT. A recipient gateway is the target of

an asset transfer, i.e., follows instructions from the source gateway. HERMES provides as strong durability guarantees as to the underlying durability guarantees of the chosen data store. If the datastore is a blockchain, HERMES can be considered to achieve transaction durability, if transactions are immutable and permanently stored in a secure decentralized ledger.

In ODAP, a client application interacts with its local gateway (source gateway GS) over a Type-1 API. The existence of this API allows the client to provide instructions to GS (corresponding to the source gateway) concerning the assets stored in the source DLT and the target DLT (via the recipient gateway, GR). It is possible that the client has complex business logic code that triggers behavior on the gateways. Hence, ODAP allows three flows: the *transfer initiation flow*, where the process is bootstrap, and several identification procedures take place; the *lock-evidence flow*, where gateways exchange proofs regarding the status of the asset to be transferred; and the *commitment establishment flow*, where the gateways commit on the asset transfer.

4.2.2 ODAP-2PC

One of the key deployment requirements of gateways for asset transfers is a high degree of gateways availability. A distributed recovery procedure then increases the resiliency of a HERMES gateway by tolerating faults. Next, we present an overview of ODAP-2PC.

The protocol is crash fault-tolerant, so the gateways are trusted to operate the ODAP protocol as specified unless they stop. We envisage ODAP-2PC to support two strategies to increase the availability of gateways [168]: (1) *self-healing mode:* after a crash, a gateway eventually recovers, informs other parties of its recovery, and continues executing the protocol; (2) *primary-backup mode:* after a crash, a gateway may never recover, but that timeout can detect this failure; if a node is crashed indefinitely, a backup is spun off, using the log storage API to retrieve the log's most recent version.

In both modes, logs are written before operations (write-ahead) to provide atomicity and consistency to the protocol used for asset exchange. The log-data is considered as resources that may be internal to the DLT system, accessible to the backup gateway and possible other gateway nodes.

There are several situations when a crash may occur. Figure 4.2 represents the crash of $\mathcal{G}_S$ before it issues a validation operation to $\mathcal{G}_R$ (steps 1 and 2). Both gateways keep their log storage APIs, with γ_{local}. For simplicity, we only represent one log storage API. In the self-healing mode, the gateway eventually recovers (step 3), building a recovered message in the form <phase, step, RECOVER, nodes> (step 4). The non-crashed gateway queries the log entries that the crashed gateway needs (steps 5, 6). In particular, $\mathcal{G}_S$ obtains the necessary log entries at step 7 and compares them to its current log. After that, $\mathcal{G}_S$ attempts to reconcile the changes with its current state (step 8). Upon processing, if both log versions match, then the log is updated, and the process can continue. If the

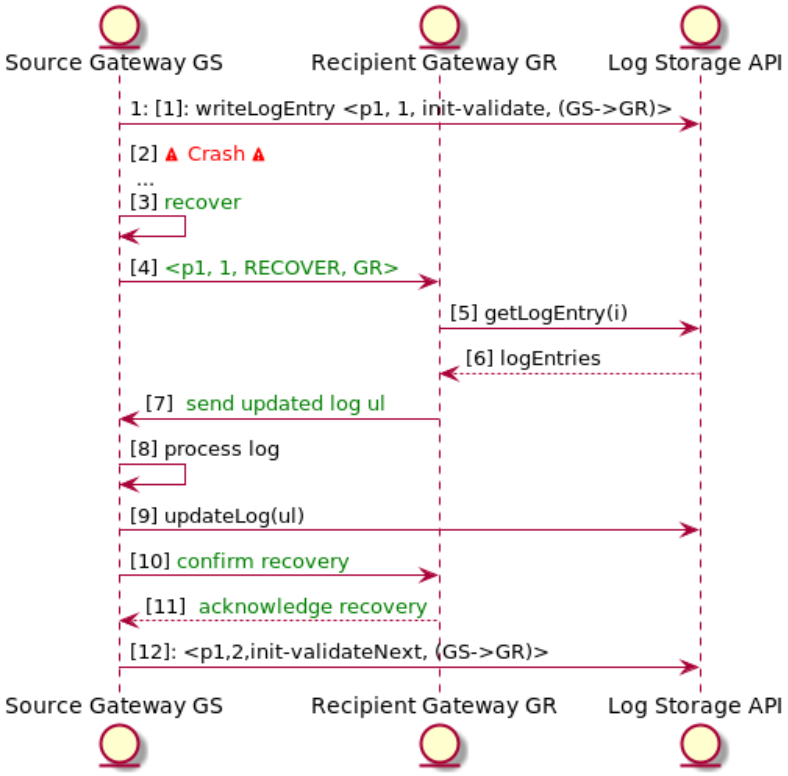

Figure 4.2: $\mathcal{G}_S$ crashing before issuing init-validation to $\mathcal{G}_R$

logs differ, then $\mathcal{G}_S$ calls the primitive `updateLog`, updating its log (step 9) and thus allowing the crashed gateway to reconstruct the current state. In this particular example, step 9 would not occur because operations exec-validate, done-validate, and ack-validate were not executed by $\mathcal{G}_R$. If the log storage API is on the shared mode, no extra steps for synchronizations are needed. After that, it confirms a successful recovery (steps 10, 11). Finally, the protocol proceeds (step 12).

Figure 4.3 represents a recovery scenario requiring further synchronization. At the retrieval of the latest log entry, $\mathcal{G}_S$ notices its log is outdated. It updates it, upon necessary validation, and then communicates its recovery to $\mathcal{G}_R$. The process then continues as normal. (for instance, corresponding to exec-validate, done-validate, and ack-validate)

ODAP-2PC is a 2PC protocol able to detect and recover from crashes, delivering the effort to execute an asset transfer starting at ODAP's phase 3: the commitment establishment flow. Crashes at other phases of the ODAP are handled by the self-healing mechanism, supported by the messaging and logging mechanism, as depicted by Figures 4.2 and 4.3. ODAP-2PC considers two parties: the coordinator $\mathcal{G}_S$, and the participant $\mathcal{G}_R$. The coordinator manages the protocol execution while the participant follows the coordinator's instructions. In phase 3, these two parties exchange sensitive messages that include the lock and unlocking of assets. Those messages may not arrive due to failures (e.g., communication failures, gateway crash due to power outage). To detect crashes, we

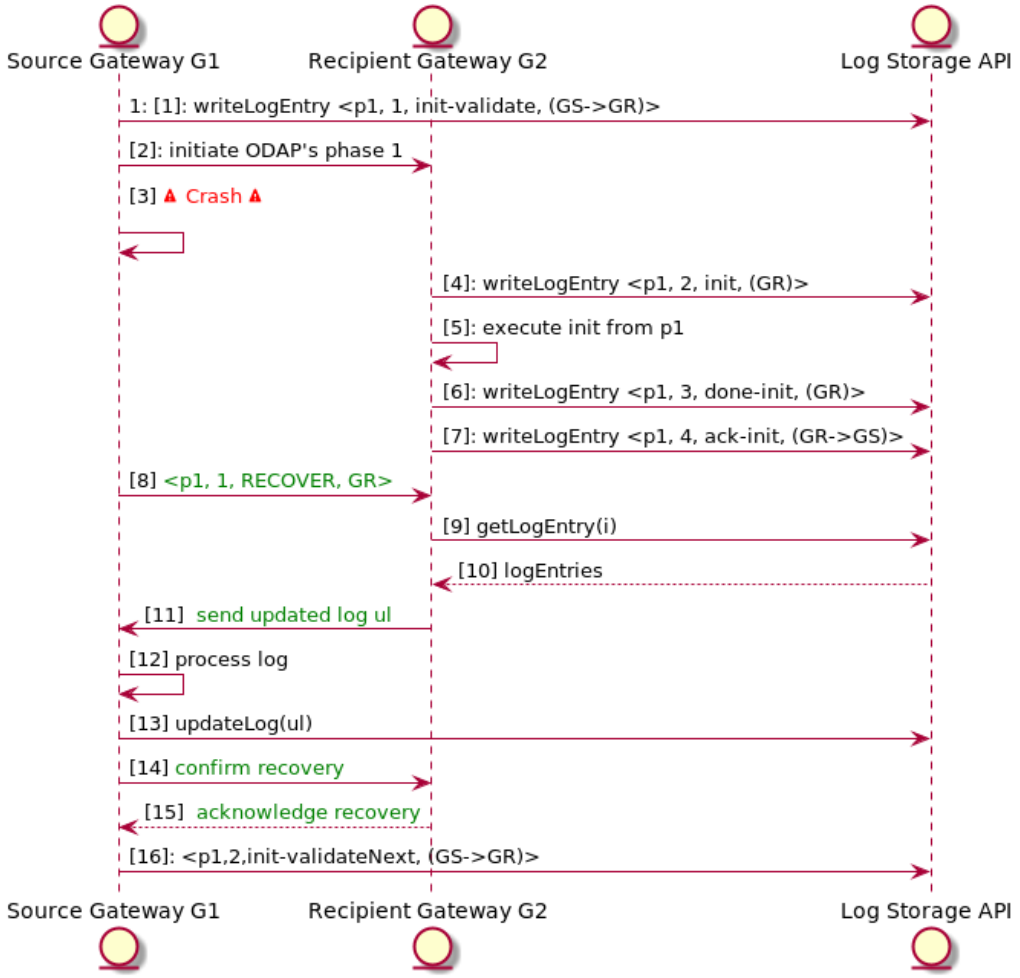

Figure 4.3: $\mathcal{G}_S$ crashing after issuing the init command to $\mathcal{G}_R$

use a timeout δ_C. However, processes may wait for the crashed gateway to recover for an unbounded timespan, wasting resources (e.g., locked assets). To avoid this, we introduce an additional timeout $\delta_{rollback}$. When a gateway does not recover before this timeout, a *timeout action* is triggered, corresponding to the *rollback protocol*. A possible rollback protocol cancels the current transactions by issuing transactions with the contrary effect, guaranteeing the consistency of the DLT whose gateway is not crashed. Upon recovery, the crashed gateway is informed of the rollback, performing a rollback too. This process guarantees the consistency of both underlying DLTs.

wudountil

Algorithm 1 depicts the ODAP-2PC. A coordinator $\mathcal{G}_S$ and a participant $\mathcal{G}_R$ perform a CC-Tx T, that typically is an asset transfer of x number of a assets, i.e., $T : \mathcal{G}_S \overset{a,x}{\rightarrow} \mathcal{G}_R$. Any time a party ABORTS, the protocol stops, and that transaction is considered invalid (and thus the run of the protocol fails). We define a set of gateway primitives Σ = {PRE_LOCK, UNLOCK, LOCK, COMMIT, CREATE_ASSET, COMPLETE, ROLLBACK}, such that they realize pre-locking an asset, locking an asset, unlocking an asset,

Algorithm 1 ODAP-2PC Protocol

Coordinator $\mathcal{G}_S$, Participant $\mathcal{G}_R$, Asset a, Gateway primitives PRE_LOCK, LOCK, COMMIT,CREATE_ASSET, COMPLETE, ROLLBACK
Asset a transferred from $\mathcal{G}_S$ to $\mathcal{G}_R$
$PO_{\mathcal{G}_S}$ = $\perp$ rollback list for $\mathcal{G}_S$
$PO_{\mathcal{G}_R}$ = $\perp$ rollback list for $\mathcal{G}_R$
▷ Pre-Voting Phase
preLock = $\mathcal{G}_S$.PRE_LOCK(a) step 2.3
$PO_{\mathcal{G}_S}$.append(preLock)
▷ Voting Phase
$\mathcal{G}_S \overset{vote-req}{\longrightarrow} \mathcal{G}_R$ step 3.1
wait until $\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S$ step 3.2
▷ Decision Phase
$\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S$ = NO $\mathcal{G}_S \overset{abort()}{\longrightarrow} \mathcal{G}_R$ otherwise, $\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S$ = YES
$\mathcal{G}_S$.ROLLBACK($PO_{\mathcal{G}_S}$) undo $\mathcal{G}_S.preLock(a)$
$lock = \mathcal{G}_S$.LOCK(a) step 3.3
$PO_{\mathcal{G}_S}$.append(lock)
$commit = \mathcal{G}_S$.COMMIT() step 3.4
$commit = \perp$
$\mathcal{G}_S \overset{abort()}{\longrightarrow} \mathcal{G}_R$
$\mathcal{G}_S.rollback(PO_{\mathcal{G}_S})$ undo $\mathcal{G}_S$.LOCK(a)
$\mathcal{G}_S \overset{commit}{\longrightarrow} \mathcal{G}_R$
$a' = \mathcal{G}_R$.CREATE_ASSET() step 3.5
$PO_{\mathcal{G}_R}$.append(a')
wait until $\mathcal{G}_R \overset{\alpha(commit)}{\longrightarrow} \mathcal{G}_S$ step 3.6
$\mathcal{G}_R \overset{\alpha(commit)}{\longrightarrow} \mathcal{G}_S$ = COMMIT$\mathcal{G}_S$.COMPLETE() step 3.8
$\mathcal{G}_S \overset{abort()}{\longrightarrow} \mathcal{G}_R$ otherwise, $\mathcal{G}_R$ failed the commit
$\mathcal{G}_S$.ROLLBACK($PO_{\mathcal{G}_S}$) undo $\mathcal{G}_S$ locks
$\mathcal{G}_R$.ROLLBACK($PO_{\mathcal{G}_R}$) undo $\mathcal{G}_R$.CREATE_ASSET()
return asset transferred

committing to a CC-Tx, creating an asset, asserting for the end of the protocol, and performing a rollback, respectively. The gateway primitives are divided into two types: off-chain primitives, and on-chain primitives, represented by $\sigma^{offchain}$ and $\sigma^{onchain}$, respectively. Some off-chain primitives call their respective on-chain primitive. The protocol receives a set of gateway primitives that realize the commit, locking, rollback and other operations. Lists $PO_{\mathcal{G}_S}$ and $PO_{\mathcal{G}_R}$ track the operations to be rolledback in case of failure for $\mathcal{G}_S$or $\mathcal{G}_R$, respectively.

First, in the session opening, the asset to be transferred is agreed on. At the pre-voting phase, the source gateway initiates the process, pre-locking an asset (executing the transaction right to the point before its commitment, at step 2.3, line 4). The recipient gateway confirms this pre-locking, issuing a VOTE-REQ to its counterparty (line 7). The recipient gateway replies either YES or ABORT (line 8), starting the decision phase. Note that the eventual ABORT, at line 8, does not require a rollback, because so far no on-chain operations took place. At the beginning of the decision phase, if $\mathcal{G}_R$ replies NO, then the pre-lock is rolledback, and the transaction aborted (lines 11 and 12). Otherwise, $\mathcal{G}_S$ tries

to lock the asset to be transferred (line 14) and commit that action (line 16). The recipient gateway completes the pending transactions (line 22) and sends an acknowledgment message back to the source gateway (line 24). Upon the second commit, the source gateway completes the process, closing the session (line 26). However, if $\mathcal{G}_S$ cannot commit (line 25 is not COMMIT), the transaction is aborted, and the respective rollbacks are triggered.

If the participant $\mathcal{G}_R$ does not reply on the blocking operations (within $t < \delta_R$, $\mathcal{G}_S$ considers $\mathcal{G}_R$ crashed, and starts the *recovery protocol*). The recovery protocol may be trivial: in ODAP-2PC, firstly, the gateway awaits for the counterparty gateway to recover (by assumption, it does). Upon recovery, the process depicted by steps 4-11 from Figure 4.2 take place. Conversely, if $\mathcal{G}_S$ does not respond within $t < \delta_S$, the same process occurs. It is worth noting that the coordinator may issue the rollback at any point $t > \delta_{rollback}$, where $\delta_{rollback} > \delta_R$, i.e., it does not need to wait indefinitely for the participant to recover. For both cases, if the recovering awaiting period is greater than the rollback timeout protocol, i.e., $t > \delta_{rollback}$, a *rollback protocol* is triggered.

4.3 Use Case: Gateway-Supported Cross-Jurisdiction Promissory Notes

In this section, we present a use case implementing digital asset transfers, benefiting from the gateway paradigm. The digital assets to be exchanged are defined as an *asset profile*, which is ongoing work at the IETF [169]. An asset profile is "the prospectus of a regulated asset that includes information and resources describing the virtual asset". A virtual asset, on its turn, is "a digital representation of value that can be digitally traded" [169]. Asset profiles can be emitted by authorized parties, having the capability to legally represent real-world assets (e.g., real estate).

4.3.1 Asset Profile

The *Asset Profile Definitions for DLT Interoperability* draft presents an unambiguous manner of representing a digital asset, independently of its concrete implementation [169]. This is notably for tokenization, as a physical asset might be represented in a multitude of ways. Thus, it is important to find a sufficiently generic schema that allows representing an arbitrary digital asset, and thus enable asset transfers. Perhaps most importantly, its definition assures that heterogenous DLTs refer to the same asset within a transfer. An asset profile contains the following fields (from [169]): issuer, asset code, asset code type, issuance date, expiration date, verification endpoint, digital signature, prospectus link, among others. We refer to this asset profile as $\mathcal{A}_p$. For generic protocols manipulating assets (e.g., transfer, creating), this asset profile can provide the necessary attributes for trust establishment. For instance, gateways should be able to verify its counter party

identity in case of an asset transfer. Moreover, the asset profile and asset code should be identifiable and retrievable, allowing different attributes to be parsed as inputs to the asset gateway primitives.

4.3.2 Using Hermes to Exchange Promissory Notes

Promissory notes are freely transferable financial instruments where issuers denote a promise to pay another party (payee) [170]. Notes are globally standardized by several legal frameworks, providing a low-risk instrument to reclaim liquidity from debt. Notes contain information regarding the debt, such as the amount, interest rate, maturity date, and issuance place. Notes are useful because they allow parties to liquidate the debts and conduct financial transactions faster, overcoming market inefficiencies. In practice, promissory notes can be both payment and credit instruments. A promissory note typically contains all the terms about the indebtedness, such as the principal amount, credit rating, interest rate, expiry date, date of issuance, and issuer's signature. Despite their benefits, paper promissory notes are hard to track, require hand signatures and not-forgery proofs, accounting for cumbersome management. To address these challenges, recent advances in promissory notes' digitalization include FQX's eNote [171]. Blockchain-supported digital promissory notes (eNotes) worth about half a million dollars were used by a "Swiss commodity trader to finance a transatlantic metal shipment" [172]. eNotes are stored in a trusted ledger covered by the legal framework, belonging to a specific jurisdiction. Consider the following supply chain scenario: a producer (P) produces a certain amount of goods that sells to a wholesaler (W). W accepted the goods, and now P issues an invoice of value V. The wholesaler could pay in, for example, 90 days. Because P does not want to wait up to 90 days for its payment, it requests a promissory note from W, stating that V will be paid in 90 days. This way, P can sell that same promissory note to a third party. The promissory note is abstract from any physical good being exchanged. Depending on the issuer, collateral might not be needed, as the accountability for liquidating the debt is tracked by the blockchain where it is stored.

Blockchain-based promissory notes belonging to a particular jurisdiction are stored in a certified blockchain that exposes a gateway. When a promissory note needs to change jurisdictions (e.g., a promissory note issued in the USA that needs to be redeemed in Europe), the gateways belonging to the source and target blockchains perform an asset transfer, where the asset is a digital promissory note. Alternatively, the gateway extends to several jurisdictions. Below is an example of an asset profile of a digital promissory note. Such digital promissory notes can be trivially exchanged between blockchains using HERMES and the ODAP-2PC protocol, where gateways belonging to different jurisdictions (e.g., representing different blockchains regulated by different entities) perform asset transfers.

4.4 Discussion

HERMES can be instantiated in blockchains supporting smart contracts that implement functionality for locking and unlocking assets. The gateway paradigm allows integrating DLT-based systems to centralized legacy systems by leveraging existing legal frameworks. For extra robustness, data integrity and counterparty performance can be attested, using trusted hardware [173, 174]. Remote attestations are particularly important, since provably exposing internal state to external parties is a crucial requirement for CC-Txs [86].

A tradeoff between reliability and performance exists. Storing logs in local storage typically has lower latency but deliver weaker integrity and availability guarantees than store them on the cloud or in a ledger. Generally, the more resilient the logging is, the higher the latency. For critical scenarios where strong accountability and traceability are needed (e.g., financial institution gateways), blockchain-based logging storage may be appropriate. Conversely, for gateways that implement interoperability between blockchains belonging to the same organization (i.e., a legal framework protects the legal entities involved), local storage might suffice.

Considering non-trusting gateways, HERMES might not be sufficiently decentralized. Besides picking the appropriate log storage support, one could choose from several techniques to decentralize gateways or to enhance the accountability level. A first option is to implement a gateway as a smart contract: this does not allow a gateway to deviate from its configured behavior but has shortcomings, such as inflexibility, lack of scalability, and operation costs. In particular, smart contracts often lack the possibility of being integrated with external resources and systems; oracles may provide some extra flexibility [22]. Smart contract-based gateways could also need to pay transaction fees in public blockchains, such as gas on Ethereum [50], raising additional costs. Additional costs imply that adding gateways on the same blockchain is not scalable. Second, to decentralize HERMES, one could implement a Byzantine fault-tolerant version of a gateway, similarly to what is planned on Cactus [175].

Regarding security, gateways should assure the integrity and non-repudiation of log entries and ensure that the protocol terminates. If an adversary performs a denial-of-service on either gateway, the asset transfer is denied but ODAP-2PC assures eventual consistency of the underlying DLTs. Accountability promoted by robust storage can diminish the impact of these attacks. The connection between gateways should always provide an authentication and authorization scheme, e.g., based on OAuth and OIDC [176], and use secure channels based on TLS/HTTPS [177].

4.5 Summary

Blockchain gateways will play an important part on interoperability between blockchains, since standardization is still needed. In this Chapter, we contributing to solve this problem with HERMES. HERMES is a middleware that enables BI across DLT-systems that can operate under different legal frameworks. HERMES is instantiated with ODAP, an asset transfer protocol between two gateway devices. HERMES supports ACID properties and can assure accountability by keeping an off-chain or on-chain shared log of operations. We propose and discuss ODAP-2PC, a distributed recovery mechanism, guaranteeing asset transfers between blockchains to be atomic and secure. By studying HERMES' reliability, performance, decentralization, security, and privacy, we explore the potential of gateways to respond to the current interoperability challenge. By presenting the digital promissory note use case, we show that HERMES is an appropriate trust anchor for enterprise use cases requiring cross-blockchain asset transfers. Future work will enable several gateways to be involved in an asset transfer (ODAP-3PC), paving the way for efficient multiparty atomic swaps.

Chapter 5

Work plan

The research work for this thesis started in December 2019. The coursework required by the doctoral program is completed apart from the course Research Seminar in Information Security I (C). We have written ten documents (7 academic papers and three technical reports). From the seven academic papers, four are accepted, and the others are submitted.

5.1 Completed Work

Our contributions not only aim to advance the state of the art in our field, but also to directly influence the industry.

The work done so far has been reported in the following documents:

1. Belchior, R., Vasconcelos, A., Guerreiro, S., & Correia, M. (2021). A Survey on Blockchain Interoperability: Past, Present, and Future Trends. ACM Computing Surveys (CSUR) 54 (8), 1-41

2. Montgomery, H., Borne-Pons, H., Hamilton, J., Bowman, M., Somogyvari, P., Fujimoto, S., Takeuchi, T., Kuhrt, T., & Belchior, R. (2020). Hyperledger Cactus Whitepaper. https://github.com/hyperledger/cactus/blob/master/docs/whitepaper/whitepaper.md

3. Belchior, R., Putz, B., Pernul, G., Correia, M., Vasconcelos, A., & Guerreiro, S. (2020). SSIBAC : Self-Sovereign Identity Based Access Control. The 3rd International Workshop on Blockchain Systems and Applications (with IEEE TrustCom 2020).

4. Belchior, R., Guerreiro, S., Vasconcelos, A., & Correia, M. (2021). A Survey on Business Process View Integration. Submitted to the Business Process Management Journal.

5. Belchior, R., Correia, M., & Hardjono, T. (2021). DLT Gateway Crash Recovery Mechanism draft 02 (Issue draft-belchior-gateway-recovery-02). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-belchior-gateway-recovery-02

6. Ghaemi, S., Rouhani, S., Belchior, R., Cruz, R. S., Khazaei, H., & Musilek, P. (2021). A Pub-Sub Architecture to Promote Blockchain Interoperability. Submitted to Computer Communications. http://arxiv.org/abs/2101.12331

7. Belchior, R., Vasconcelos, A., Correia, M., & Hardjono, T. (2021). HERMES: Fault-Tolerant Middleware for Blockchain Interoperability. Submitted to Future Generation Computer Systems.

8. Hargreaves, M., Hardjono, T., & Belchior, R. (2021). Open Digital Asset Protocol draft 02 (Issue draft-hargreaves-odap-02). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-hargreaves-odap-02

9. Belchior, R., Vasconcelos, A., Correia, M., & Hardjono, T. (2021). Enabling Cross-Jurisdiction Digital Asset Transfer. IEEE International Conference on Services Computing.

The following publications are done within the duration of the PhD, but not directly related to it:

1. Rouhani, S., Belchior, R., Cruz, R. S., & Deters, R. (2021). Distributed attribute-based access control system using permissioned blockchain. World Wide Web. https://doi.org/10.1007/s11280-021-00874-7

2. Belchior, Rafael; Correia, Miguel; and Vasconcelos, André, "Towards Secure, Decentralized, and Automatic Audits With Blockchain" (2020). In Proceedings of the 28th European Conference on Information Systems (ECIS), An Online AIS Conference, June 15-17, 2020.

5.2 Future Work

For future work, we plan on finishing and submitting our on-going work. Table 5.1 shows completed work and future work. This table contains only peer-reviewed publications and technical reports, excluding other forms of dissemination, such as invited speeches, tutoring and teaching activities, scientific meetings, and symposiums.

The plan for future work is as follows:

Accepted and under revision publications per research objective	State (August 2021)	Target Quarter
Objective 1		
P1.1: A Survey on Blockchain Interoperability: Past, Present, and Future Trends	✓	-
P1.2: Do you need a Distributed Ledger Technology interoperability solution?	⊙	2021, Q4
Objective 2		
P2.1: A Survey on Business Process View Integration	✓	-
P2.2: SSIBAC: Self-Sovereign Identity Based Access Control	✓	-
P2.3: BUNGEE: Visualizing, Merging, and Processing Blockchain Views	⊙	-
P2.4: Linking the Chains:Analysis and Visualization of Cross-Chain Transactions	⊙	2022, Q2
Objective 3		
P3.1: Hyperledger Cactus Whitepaper	✓	-
P3.2: DLT Gateway Crash Recovery Mechanism	✓	-
P3.3: Open Digital Asset Protocol	✓	-
P3.4: A Pub-Sub Architecture to Promote Blockchain Interoperability	✓	-
P3.5: Enabling Cross-Jurisdiction Digital Asset Transfer	✓	-
P3.6: HERMES: Fault-Tolerant Middleware for Blockchain Interoperability	✓	-
P3.7: Cross-Chain Transfer With the Open Digital Asset Protocol	⊙	2022, Q2
P3.8 Hyperledger Cactus: A Distributed Operating System Enabling Blockchain Interoperability	⊙	2022, Q1
P3.9: Ares: *Decentralized, Fault-Tolerant Blockchain Migrators*	⊙	2022, Q3

Table 5.1: State of the publications made in the context of this dissertation on August 2021, per research objective. The state of publications is either accepted or under review, i.e., completed (represented by ✓) or work in progress, (represented by ⊙).

2021, Q4

Do you need a Distributed Ledger Technology interoperability solution?: this paper completes our survey on blockchain interoperability. First, we aim to introduce an updated classification framework, based on existing design patterns for blockchain interoperability solutions. Next, we aim to derive a framework for a developer to choose the right interoperability solution, based on a set of requirements and constraints. We already started writing this paper.

BUNGEE: Visualizing, Merging, and Processing Blockchain Views: data portability solution, where a blockchain view can represent stakeholder-specific views. This is the basis for data migration. Data migrations have been performed before on public blockchains [22] to render flexibility to blockchain-based solutions. Such work proposes data migration capabilities and patterns for public, permissionless blockchains, in which a user can specify requirements and scope for the blockchain infrastructure supporting their service. However, automatic smart contract migrations have not been sufficiently explored up to this date. This objective aims to implement a use case of a complete blockchain migration (data and smart contract migration). The underlying research questions are *How to migrate data, functionality, and permissions of a DLT-based application to another DLT*, and *How to assure safety and liveness of the migration process?* We already started writing this paper.

2022, Q1

Hyperledger Cactus: A Distributed Operating System Enabling Blockchain Interoperability: this paper provides a universal composability model for blockchain interoperability. Introduce a formalization of Hyperledger Cactus.

2022, Q2

Cross-Chain Transfer With the Open Digital Asset Protocol: a paper continuing Hermes, exploring the practical implementation of ODAP. This objective aims to perform legal, standardized asset transfers across DLTs with different jurisdictions. The underlying research questions are: *What are the necessary requirements for an infrastructure conducting asset transfers with legal binding across DLTs?*

Linking the Chains:Analysis and Visualization of Cross-Chain Transactions: in this paper we study, design, and implement a supporting tool that aggregates data from several blockchains, parses it, and exposes it to an user interface.

2022, Q3

ARES: *Decentralized, Fault-Tolerant Blockchain Migrators*: a use case migrating assets and smart contracts across blockchains.

Create a working group at IETF and submit the drafts on ODAP and Crash Recovery as RFCs

2022, Q4

Write and submit the PhD thesis.

Finally, we present the completion of the thesis' objectives and the key publications supporting it in Table 5.2.

As shown in Table 5.1, the first objective is partially completed. We have explored the available blockchain solutions in P1.1, completing Objective 1.2. Building on that knowledge, we aim to complete Objectives 1.1 and 1.3 by studying the technical requirements for a general-purpose blockchain interoperability solution (P3.8) and derive a decision model to choose a blockchain interoperability solution (P1.2), respectively.

Objective 1 addresses the need for the contributions of Objective 2. We have already completed Objective 2.2 by delivering SSIBAC. SSIBAC can support authenticating entities manipulating blockchain views and using tools for cross-chain analysis (Objectives 2.1 and 2.3). The key enablers will be integral parts of the hybrid connectors. In particular, some hybrid connectors might have tools to analyze cross-chain transactions, utilize blockchain views (e.g., for migrating data), and the SSIBAC for authorization.

Objective	Status	Publications	Description
Objective 1.1	±	P1.2, P3.8	Technical requirements
Objective 1.2	✓	P1.1	Available solutions
Objective 1.3	±	P1.2	Decision model
Objective 2.1	±	P2.1,P2.3	Data portability
Objective 2.2	✓	P2.2	Identity portability
Objective 2.3	✗	P2.4	Tools for analysis
Objective 3	±	P3.1-P3.9	Hybrid Connectors

Table 5.2: State of the completion of the objectives of this thesis (August 2021). Objectives are completed (represented by ✓), partially completed or work in progress (i.e., needs more supporting work, represented by ±), or not started (represented by ✗). Publications refer Table 5.1.

The first two objectives allow the fulfillment of the third objective. We already have preliminary results on implementing hybrid connectors (Objective 3). However, the knowledge of the unfinished objectives will support designing and implementing reliable decentralized hybrid connectors and centralized hybrid connectors with more capabilities. We are currently focusing on writing documents P1.2 and P3.8. The combination of these objectives addresses the end goal of this doctoral work.

5.3 Other Collaborations

Counting with almost 200 stars and more than 100 forks, Hyperledger Cactus [175] is the most popular project dedicated to enterprise interoperability, backed by Hyperledger, Accenture, and Fujitsu. The code of most of our contributions is incorporated into the main codebase of Cactus, being available for researchers and practitioners alike.

On the other hand, our work on Hermes yielded several drafts in the context of a forming working group at the Internet Engineering Task Force, a standardization organization responsible for TLS [177]. Two IETF drafts are the direct outcome of our work: the ODAP draft [166] and the Crash Recovery draft [168]. We submit updated versions of the draft in collaboration with MIT Connection Science on a Github repository[1]. We hope that our standardization effort, which takes both academics and practitioners, can yield one or more RFCs[2], paving the way for standardization in the space.

We collaborated with the Linux Foundation on blockchain research via the Hyperledger Foundation, namely:

- The Hyperledger Fabric Based Access Control project[3]: this project yielded our pa-

[1] https://github.com/CxSci/blockchain-gateway
[2] https://www.ietf.org/standards/rfcs/
[3] https://wiki.hyperledger.org/display/INTERN/Hyperledger+Fabric+Based+Access+Control

per *Distributed attribute-based access control system using permissioned blockchain* [178], published in the journal World Wide Web.

- The Towards Blockchain Interoperability with Hyperledger[4]: this project yield our paper *A Pub-Sub Architecture to Promote Blockchain Interoperability* [179], currently submitted at the journal Future Generation Computing Systems.
- The Visualization and Analysis of Cross-chain Transactions[5]: this project is ongoing.
- The Cactus-samples - Business Logic Plugins for Hyperledger Cactus project [6]: this project is ongoing.

[4]https://wiki.hyperledger.org/display/INTERN/Towards+Blockchain+Interoperability+with+Hyperledger

[5]https://wiki.hyperledger.org/display/INTERN/Visualization+and+Analysis+of+Cross-chain+Transactions

[6]https://wiki.hyperledger.org/display/INTERN/Cactus-samples+-+Business+Logic+Plugins+for+Hyperledger+Cactus

Chapter 6

Conclusion

This document presented the status of the doctoral research. The work developed so far partially tackles each of the three research objectives of this thesis: study the status quo of blockchain interoperability, develop supporting technologies and tools supporting blockchain interoperability, and create blockchain interoperability solutions suited for enterprise systems.

In this work, we presented our framework to systematically compare blockchain interoperability solutions, the blockchain interoperability framework. After that, we presented Hermes, a hybrid connector. In particular, Hermes, powered by ODAP, can perform cross-jurisdiction asset transfers, paving the way for standardized cross-jurisdiction asset transfers. Gateway could support self-sovereign-based access control (with SSIBAC). SSIBAC is a privacy-friendly access control model that promotes identity portability.

For future work, we plan on: 1) define technical requirements and a general model for blockchain interoperability, 2) build data portability and visualization tools for hybrid connectors, and 3) implement a hybrid connector. We plan to validate our hybrid connector with the blockchain migration and digital asset cross-chain transfer use cases.

Bibliography

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

[2] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).
URL http://bitcoin.org/bitcoin.pdf

[3] A. Zohar, Bitcoin: Under the Hood, Communications of the ACM 58 (9) (2015) 104–113.

[4] F. Ul Hassan, A. Ali, S. Latif, J. Qadir, S. Kanhere, J. Singh, J. Crowcroft, Blockchain And The Future of the Internet: A Comprehensive Review, arXiv e-prints 1904.00733 (2019). arXiv:arXiv:1904.00733.
URL https://arxiv.org/abs/1904.00733

[5] R. Belchior, M. Correia, A. Vasconcelos, JusticeChain: Using Blockchain To Protect Justice Logs, in: CoopIS 2019: 27th International Conference on Cooperative Information Systems, 2019.

[6] R. Belchior, A. Vasconcelos, M. Correia, Towards Secure, Decentralized, and Automatic Audits with Blockchain, in: European Conference on Information Systems, 2020.

[7] S. Rouhani, R. Deters, Blockchain based access control systems: State of the art and challenges, in: Proceedings - 2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, 2019. doi:10.1145/3350546.3352561.

[8] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, S. Guerreiro, SSIBAC : Self-Sovereign Identity Based Access Control, in: The 3rd International Workshop on Blockchain Systems and Applications, IEEE, 2020.

[9] F. Casino, T. K. Dasaklis, C. Patsakis, A systematic literature review of blockchain-based applications: Current status, classification and open issues, Telematics and Informatics 36 (2019) 55–81. doi:10.1016/j.tele.2018.11.006.

[10] L. Pawczuk, M. Gogh, N. Hewett, 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).
URL www.weforum.org

[11] T. Hardjono, Blockchain Gateways, Bridges and Delegated Hash-Locks, arXiv 2102.03933 (2021). arXiv:2102.03933.
URL http://arxiv.org/abs/2102.03933

[12] N. Atzei, M. Bartoletti, T. Cimoli, A Survey of Attacks on Ethereum Smart Contracts, in: International Conference on Principles of Security and Trust, Association for Computing Machinery, 2017, pp. 164–186.

[13] E. Anceaume, A. Del Pozzo, R. Ludinard, M. Potop-Butucaru, S. Tucci-Piergiovanni, Blockchain Abstract Data Type (feb 2018). arXiv:1802.09877.
URL http://arxiv.org/abs/1802.09877

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

[15] L. M. Maruping, V. Venkatesh, R. Agarwal, A control theory perspective on agile methodology use and changing user requirements, Information Systems Research 20 (3) (2009) 377–399.

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

[17] T. Hardjono, A. Lipton, A. Pentland, Toward an interoperability architecture for blockchain autonomous systems, IEEE Transactions on Engineering Management (2019).

[18] H. Tam Vo, Z. Wang, D. Karunamoorthy, J. Wagner, E. Abebe, M. Mohania, Internet of Blockchains: Techniques and Challenges Ahead, in: 2018 IEEE iThings/GreenCom/CPSCom/SmartData, 2018, pp. 1574–1581.

[19] National Interoperability Framework Observatory, European Interoperability Framework (2020).
URL https://joinup.ec.europa.eu/collection/nifo-national-interoperability-framewo
3-interoperability-layers#3.6

[20] E. Fynn, F. Pedone, B. Alysson, Smart Contracts on the Move, in: Dependable Systems and Networks, 2020.

[21] G. Wang, Z. Jerry, M. Nixon, SoK : Sharding on Blockchain, in: ACM Conference on Advances in Financial Technologies, 2019.

[22] R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, A Survey on Blockchain Interoperability: Past, Present, and Future Trends, ACM Computing Surveys (may 2021). arXiv:2005.14282.
URL http://arxiv.org/abs/2005.14282

[23] DeFi Report Q2 2021 — Consensys.
URL https://consensys.net/reports/defi-report-q2-2021

[24] W. B. Group, Blockchain Interoperability (2020).
URL https://www.ft.com/content/1cfb6d46-5d5a-11e9-939a-341f5ada9d40

[25] A. Lipton, Cryptocurrencies change everything, Quantitative Finance 21 (8) (2021) 1257–1262. doi:10.1080/14697688.2021.1944490.
URL https://www.tandfonline.com/doi/abs/10.1080/14697688.2021.1944490

[26] Global Agenda Council on the Future of Software & Society Deep Shift Technology Tipping Points and Societal Impact (2015).

[27] Deloitte's 2019 Global Blockchain Survey.

[28] Hype Cycle for Blockchain, 2021.
URL https://www.gartner.com/en/documents/4003463/hype-cycle-for-blockchain-2021

[29] J. Kolb, M. Abdelbaky, R. H. Katz, D. E. Culler, Core concepts, challenges, and future directions in blockchain: A centralized tutorial, ACM Computing Surveys 53 (1) (2 2020). doi:10.1145/3366370.

[30] M. Riddley, Amara's Law — Matt Ridley (2017).
URL https://www.rationaloptimist.com/blog/amaras-law/

[31] R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, A Survey on Blockchain Interoperability: Past, Present, and Future Trends, ACM Computing Surveys (5 2021).
URL http://arxiv.org/abs/2005.14282

[32] Inclusive Deployment of Blockchain for Supply Chains: Part 6-A Framework for Blockchain Interoperability In Collaboration with Deloitte, Tech. rep. (2020).
URL www.weforum.org

[33] Bridging the Governance Gap: Interoperability for blockchain and legacy systems, Tech. rep. (2020).

[34] D. Chen, G. Doumeingts, F. Vernadat, Architectures for enterprise integration and interoperability: Past, present and future, Computers in Industry 59 (7) (2008) 647–659. doi:10.1016/J.COMPIND.2007.12.016.

[35] H. Sun, H. Mao, X. Bai, Z. Chen, K. Hu, W. Yu, Multi-blockchain model for central bank digital currency, in: Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings, Vol. 2017-Decem, IEEE Computer Society, 2018, pp. 360–367. doi:10.1109/PDCAT.2017.00066.

[36] M. Christodorescu, W. C. Gu, R. Kumaresan, M. Minaei, M. Ozdayi, B. Price, S. Raghuraman, M. Saad, C. Sheffield, M. Xu, M. Zamani, Towards a Two-Tier Hierarchical Infrastructure: An Offline Payment System for Central Bank Digital Currencies, arXiv 2012.08003 (2020) 1–21arXiv:2012.08003.
URL http://arxiv.org/abs/2012.08003

[37] A. Sardon, T. Hardjono, Blockchain Gateways: Use-Cases (draft-sardon-blockchain-gateways-usecases-00) (2020).
URL https://datatracker.ietf.org/doc/draft-sardon-blockchain-gateways-usecases/

[38] H. Montgomery, H. Borne-Pons, J. Hamilton, M. Bowman, P. Somogyvari, S. Fujimoto, T. Takeuchi, T. Kuhrt, R. Belchior, Hyperledger Cactus Whitepaper (2020).
URL https://github.com/hyperledger/cactus/blob/master/docs/whitepaper/whitepaper.md

[39] R. Belchior, S. Guerreiro, A. Vasconcelos, M. Correia, A Survey on Business Process View Integration (nov 2020). arXiv:2011.14465.
URL http://arxiv.org/abs/2011.14465

[40] Digital Asset, DAML SDK 1.1.1 documentation (2019).
URL https://docs.daml.com/getting-started/installation.html

[41] Quant Foundation, Overledger Network Whitepaper v0.3, Tech. rep., Quant (2019).

[42] R. Belchior, A. Vasconcelos, M. Correia, T. Hardjono, Enabling Cross-Jurisdiction Digital Asset Transfer, in: IEEE International Conference on Services Computing, IEEE, 2021.

[43] C. Cachin, M. Vukolić, Blockchain Consensus Protocols in the Wild, arXiv e-prints 91 (jul 2017). arXiv:1707.01873.
URL http://arxiv.org/abs/1707.01873

[44] N. Szabo, Formalizing and securing relationships on public networks, First Monday 2 (9) (1997).

[45] V. Buterin, Ethereum: A next-generation smart contract and decentralized application platform (2014).
URL https://github.com/ethereum/wiki/wiki/White-Paper

[46] A. M. Antonopoulos, G. Wood, Mastering Ethereum: building smart contracts and dapps, O'Reilly Media, 2018.

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

[48] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends, in: Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 557–564.

[49] J. Kwon, E. Buchman, Cosmos Whitepaper, Tech. rep., Cosmos Foundation (2016).

[50] G. Wood, Ethereum: A secure decentralised generalised transaction ledger. Byzantium version 7e819ec, Tech. rep. (2019).
URL https://ethereum.github.io/yellowpaper/paper.pdf

[51] R. Brown, The Corda Platform: An Introduction White Paper (2018).
URL https://www.r3.com/reports/the-corda-platform-an-introduction-whitepaper/

[52] JP Morgan, Quorum White Paper (2017).
URL https://github.com/jpmorganchase/quorum/blob/master/docs/QuorumWhitepaperv0.2.pdf

[53] G. Greenspan, MultiChain White Paper (2015).
URL https://www.multichain.com/white-paper/

[54] L. Kan, Y. Wei, A. Hafiz Muhammad, W. Siyuan, G. Linchao, H. Kai, A Multiple Blockchains Architecture on Inter-Blockchain Communication, Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018 (2018) 139–145.

[55] M. Conti, K. E. Sandeep, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, IEEE Communications Surveys and Tutorials 20 (4) (2018) 3416–3452.

[56] M. Hargreaves, R. Hardjono, Thomas Belchior, Open Digital Asset Protocol (draft-hargreaves-odap-02) (2021).
URL https://datatracker.ietf.org/doc/draft-hargreaves-odap/

[57] R. Belchior, A. Vasconcelos, M. Correia, T. Hardjono, HERMES: Fault-Tolerant Middleware for Blockchain Interoperability, TechrXiv 14120291/1 (mar 2021). arXiv:1, doi:10.36227/TECHRXIV.14120291.V1.
URL /articles/preprint/HERMES_Fault-Tolerant_Middleware_for_Blockchain_Interoperability/14120291/1

[58] A. Garoffolo, D. Kaidalov, R. Oliynykov, Zendoo: a zk-SNARK Verifiable Cross-Chain Transfer Protocol Enabling Decoupled and Decentralized Sidechains, Tech. rep., V.N.Karazin Kharkiv National University (2020).

[59] Interledger, Interledger Protocol V4 (ILPv4) — Interledger (2020).
URL https://interledger.org/rfcs/0027-interledger-protocol-4/

[60] Kyber Network, Peace Relay (2018).
URL https://github.com/KyberNetwork/peace-relay

[61] Ethereum Foundation, Consensys, BTC-relay: Ethereum contract for Bitcoin SPV (2015).
URL https://github.com/ethereum/btcrelay

[62] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, W. J. Knottenbelt, SoK: Communication Across Distributed Ledgers, Tech. rep. (2019).
URL https://eprint.iacr.org/2019/1128.pdf

[63] N. Asokan, V. Shoup, M. Waidner, Optimistic fair exchange of digital signatures, in: International Conference on the Theory and Applications of Cryptographic Techniques, Vol. 1403, Springer Verlag, 1998, pp. 591–606.

[64] M. Borkowski, C. Ritzer, D. Mcdonald, S. Schulte, Caught in Chains: Claim-First Transactions for Cross-Blockchain Asset Transfers (2018).
URL http://www.infosys.tuwien.ac.at/tast/

[65] P. Lafourcade, M. Lombard-Platet, About blockchain interoperability, Information Processing Letters 161 (2020) 105976.

[66] F. B. Vernadat, Interoperable enterprise systems: Architectures and methods, in: IFAC Proceedings Volumes (IFAC-PapersOnline), Vol. 12, Elsevier, 2006, pp. 13–20.

[67] B. Pillai, K. Biswas, Blockchain Interoperable Digital Objects, in: ICBC2019 International Conference on Blockchain, 2019.
URL https://doi.org/10.1007/978-3-030-23404-1_6

[68] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain Technology Overview, Tech. rep., NISTIR (2018). doi:02.
URL https://nvlpubs.nist.gov/

[69] L. Besançon, C. Silva, P. Ghodous, Towards Blockchain Interoperability: Improving Video Games Data Exchange, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency, 2019.

[70] H. Jin, X. Dai, Towards a novel architecture for enabling interoperability amongst multiple blockchains, in: IEEE 38th International Conference on Distributed Computing Systems, 2018.

[71] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, Y. C. Hu, Hyperservice: Interoperability and programmability across heterogeneous blockchains, in: Proceedings of the ACM Conference on Computer and Communications Security, 2019, pp. 549–566. arXiv:1908.09343, doi:10.1145/3319535.3355503.
URL https://dl.acm.org/doi/abs/10.1145/3319535.3355503

[72] G. Birkhoff, Lattice Theory, Volume 25, Part 2, American Mathematical Soc., 1940.
URL https://books.google.com/books?id=0Y8d-MdtVwkC&pgis=1

[73] R. Barnes, Factors in the Portability of Tokenized Assets on Distributed Ledgers, arXiv pre-prints (5 2020).
URL http://arxiv.org/abs/2005.07461

[74] B. Pillai, K. Biswas, V. Muthukkumarasamy, Cross-chain interoperability among blockchain-based systems using transactions, Knowledge Engineering Review 35 (2020) 1–18. doi:10.1017/S0269888920000314.

[75] Ethereum Foundation, ETH 2 Phase 2 WIKI (2019).
URL https://hackmd.io/UzysWse1Th240HELswKqVA?view

[76] J. Chen, S. Micali, Algorand (2016) 51–68.
URL http://arxiv.org/abs/1607.01341

[77] F. Vogelsteller, V. Buterin, EIP 20: ERC-20 Token Standard (2015).
URL https://eips.ethereum.org/EIPS/eip-20

[78] Security Token Standard, SecurityTokenStandard/EIP-Spec (2019).
URL https://github.com/SecurityTokenStandard/EIP-Spec

[79] G. Medcraft, Regulatory Approaches to the Tokenisation of Assets, Tech. rep., OECD (2021).
URL https://www.oecd.org/finance/regulatory-approaches-to-the-tokenisation-of-ass
htm

[80] Swiss Financial Market Supervisory Authority, FINMA publishes ICO guidelines (2018).
URL https://www.finma.ch/en/news/2018/02/20180216-mm-ico-wegleitung/

[81] U.S. Securities and Exchange Commission, Framework for "Investment Contract" Analysis of Digital Assets 1, Tech. rep. (2019).
URL https://www.sec.gov/files/dlt-framework.pdf

[82] P. Gaži, A. Kiayias, D. Zindros, Proof-of-stake sidechains, IEEE Symposium on Security and Privacy (2019).

[83] P. Frauenthaler, M. Borkowski, S. Schulte, A Framework for Blockchain Interoperability and Runtime Selection, arXiv preprint (5 2019).
URL https://arxiv.org/abs/1905.07014

[84] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, P. Wuille, Enabling Blockchain Innovations with Pegged Sidechains, Tech. rep., Blockstream (2014).

[85] S. Schulte, M. Sigwart, P. Frauenthaler, M. Borkowski, Towards Blockchain Interoperability, in: International Conference on Business Process Management: BPM 2019: Business Process Management: Blockchain and Central and Eastern Europe Forum, Vol. 361, Springer Verlag, 2019, pp. 3–10. doi:10.1007/978-3-030-30429-4{_}1.
URL https://www.dsg.tuwien.ac.at

[86] E. Abebe, D. Karunamoorthy, J. Yu, Y. Hu, V. Pandit, A. Irvin, V. Ramakrishna, Verifiable Observation of Permissioned Ledgers, arXiv 2012.07339v2 (2021). arXiv:2012.07339v2.

[87] J. Poon, T. Dryja, The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, Tech. rep., Lightning Network (2016).
URL https://lightning.network/lightning-network-paper.pdf

[88] M. Hargreaves, T. Hardjono, Open Digital Asset Protocol (draft-hargreaves-odap-01) (2020).
URL https://datatracker.ietf.org/doc/draft-hargreaves-odap/

[89] R. Belchior, M. Correia, T. Hardjono, DLT Gateway Crash Recovery Mechanism (draft-belchior-gateway-recovery-00), Tech. rep. (2021).
URL https://datatracker.ietf.org/doc/draft-belchior-gateway-recovery/

[90] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, W. J. Knottenbelt, XCLAIM: A Framework for Blockchain Interoperability, in: IEEE Symposium on Security & Privacy, 2019.

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

[92] V. Buterin, An Incomplete Guide to Rollups (2021).
URL https://vitalik.ca/general/2021/01/05/rollup.html

[93] P. Robinson, R. Ramesh, General Purpose Atomic Crosschain Transactions, arXiv (11 2020).
URL http://arxiv.org/abs/2011.12783

[94] G. Myers, T. Badgett, C. Sandler, Test-Case Design, in: The Art of Software Testing, John Wiley \& Sons, Ltd, 2012, Ch. 4, pp. 41–84. doi:https://doi.org/10.1002/9781119202486.ch4.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119202486.ch4

[95] T. Durieux, J. F. Ferreira, R. Abreu, P. Cruz, Empirical review of automated analysis tools on 47,587 ethereum smart contracts, in: Proceedings - International Conference on Software Engineering, IEEE Computer Society, 2020, pp. 530–541.

[96] J. Poon, V. Buterin, Plasma: Scalable Autonomous Smart Contracts, Tech. rep., Plasma (2017).
URL https://plasma.io/

[97] H. Wang, D. He, X. Wang, C. Xu, W. Qiu, Y. Yao, Q. Wang, An Electricity Cross-Chain Platform Based on Sidechain Relay, in: Journal of Physics: Conference Series, Vol. 1631, IOP Publishing Ltd, 2020, p. 12189. doi:10.1088/1742-6596/1631/1/012189.
URL https://iopscience.iop.org/article/10.1088/1742-6596/1631/1/012189https://iopscience.iop.org/article/10.1088/1742-6596/1631/1/012189/meta

[98] P. Frauenthaler, M. Sigwart, C. Spanring, S. Schulte, Testimonium : A Cost-Efficient Blockchain Relay, arXiv Preprints (2020).

[99] T. Baneth, Waterloo — a Decentralized Practical Bridge between EOS and Ethereum (2019).
URL https://blog.kyber.network/waterloo-a-decentralized-practical-bridge-between-

[100] P. Frauenthaler, M. Sigwart, C. Spanring, M. Sober, S. Schulte, ETH Relay: A Cost-efficient Relay for Ethereum-based Blockchains, in: 2020 IEEE International Conference on Blockchain (Blockchain), IEEE, 2020, pp. 204–213. doi:10.1109/Blockchain50366.2020.00032.
URL https://ieeexplore.ieee.org/document/9284781/

[101] V. Arasev, POA Network Whitepaper, Tech. rep., POA Network (2017).
URL https://www.poa.network/for-users/whitepaper

[102] A. Jain, P. Schilz, Block Collider Whitepaper, Tech. rep. (2017).
URL https://www.blockcollider.org/whitepaper

[103] Loom, Intro to Loom Network — Loom SDK (2016).
URL https://loomx.io/developers/en/intro-to-loom.html

[104] J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, M. Friedenbach, Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks, Tech. rep., BlockStream (2016).
URL http://arxiv.org/abs/1612.05491

[105] A. Poelstra, A. Back, M. Friedenbach, G. Maxwell, P. W. Blockstream, Blockstream: Confidential Assets, Tech. rep. (2017).
URL https://blockstream.com/bitcoin17-final41.pdf

[106] R. Khalil, P. Moreno-Sanchez, A. Zamyatin, A. Gervais, G. Felley, Commit-Chains: Secure, Scalable Off-Chain Payments, Tech. rep.
URL https://github.com/liquidity-network/nocust-contracts-solidity

[107] A. Culwick, D. Metcalf, Blocknet design specification v1.0, Tech. rep., Blocknet (2018).
URL https://www.blocknet.co/wp-content/uploads/whitepaper/Blocknet_Whitepaper.pdf

[108] S. Lerner, RSK Whitepaper, Tech. rep., RSK (2015).
URL https://docs.rsk.co/RSK_White_Paper-Overview.pdf

[109] R. Lan, G. Upadhyaya, S. Tse, M. Zamani, Horizon: A Gas-Efficient, Trustless Bridge for Cross-Chain Transactions (1 2021).
URL http://arxiv.org/abs/2101.06000

[110] O. Shlomovits, O. Leiba, JugglingSwap: Scriptless Atomic Cross-Chain Swaps, arXiv (7 2020).
URL http://arxiv.org/abs/2007.14423

[111] N. Shadab, F. Hooshmand, M. Lesani, Cross-chain Transactions, in: IEEE International Conference on Blockchain and Cryptocurrency, 2020.

[112] M. Belotti, S. Moretti, M. Potop-Butucaru, S. Secci, Game Theoretical Analysis of Atomic Cross-Chain Swaps, Hal Archives-Ouverte hal-02414356 (2020).
URL https://hal.archives-ouvertes.fr/hal-02414356

[113] P. Robinson, D. Hyland-Wood, R. Saltini, S. Johnson, J. Brainard, Atomic Crosschain Transactions for Ethereum Private Sidechains, Tech. rep. (2019).

[114] A. Deshpande, M. Herlihy, Privacy-preserving cross-chain atomic swaps, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 12063 LNCS, Springer, 2020, pp. 540–549. doi:10.1007/978-3-030-54455-3{_}38.
URL https://link.springer.com/chapter/10.1007/978-3-030-54455-3_38

[115] J. Gugger, Bitcoin-Monero Cross-chain Atomic Swap, Cryptology ePrint Archive (2020).
URL https://eprint.iacr.org/2020/1126

[116] Hyperledger, Hyperledger Quilt Documentation (2019).
URL https://wiki.hyperledger.org/display/quilt/Hyperledger+Quilt

[117] W. Warren, A. Bandeali, 0x: An open protocol for decentralized exchange on the Ethereum blockchain, Tech. rep. (2017).

[118] Mayer Christoph, Mai Jesse N, Tom M, Tokrex Whitepaper, Tech. rep., Tokrex (2017).
URL www.tokrex.org

[119] H. Tian, K. Xue, S. Li, J. Xu, J. Liu, J. Zhao, Enabling Cross-chain Transactions: A Decentralized Cryptocurrency Exchange Protocol, arXiv (5 2020).
URL http://arxiv.org/abs/2005.03199

[120] J. Lu, B. Yang, Z. Liang, Y. Zhang, S. Demmon, E. Swartz, L. Lu, Wanchain: Building Super Financial Markets for the New Digital Economy (2017).
URL https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf

[121] COMIT, COMIT Protocol (2020).
URL DisincentivizingDoubleSpendAttacksAcrossInteroperableBlockchains.

[122] Fusion Foundation, An Inclusive Cryptofinance Platform Based on Blockchain, Tech. rep., Fusion Foundation (2017).

[123] B. Dai, S. Jiang, M. Zhu, M. Lu, D. Li, C. Li, Research and Implementation of Cross-Chain Transaction Model Based on Improved Hash-Locking, in: Communications in Computer and Information Science, Vol. 1267, Springer Science and Business Media Deutschland GmbH, 2020, pp. 218–230. doi:10.1007/978-981-15-9213-3{_}17.
URL https://link.springer.com/chapter/10.1007/978-981-15-9213-3_17

[124] J. Rueegger, G. S. MacHado, Rational Exchange: Incentives in Atomic Cross Chain Swaps, in: IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020, Institute of Electrical and Electronics Engineers Inc., 2020. doi:10.1109/ICBC48266.2020.9169408.

[125] G. Wood, Polkadot: Vision for a Heterogeneous Multi-Chain Framework, Tech. rep. (2016).
URL https://github.com/w3f/polkadot-white-paper/raw/master/PolkaDotPaper.pdf

[126] Komodo, Komodo Whitepaper, Tech. rep., Komodo (2018).
URL https://komodoplatform.com/wp-content/uploads/2018/06/Komodo-Whitepaper-June-3.pdf

[127] ARK, ARK Whitepaper Version 2.1.0 (2019).
URL https://whitepaper.ark.io/prologue

[128] M. Spoke, Aion: Enabling the decentralized Internet, Tech. rep. (2017).
URL https://whitepaper.io/document/31/aion-whitepaper

[129] M. Nissl, E. Sallinger, S. Schulte, M. Borkowski, Towards Cross-Blockchain Smart Contracts (10 2020).
URL http://arxiv.org/abs/2010.07352

[130] G. Falazi, U. Breitenbücher, F. Daniel, A. Lamparelli, F. Leymann, V. Yussupov, Smart Contract Invocation Protocol (SCIP): A Protocol for the Uniform Integration of Heterogeneous Blockchain Smart Contracts, in: International Conference on Advanced Information Systems Engineering, Vol. 12127 LNCS, 2020, pp. 134–149.

[131] G. Bu, R. Haouara, T. S. L. Nguyen, M. Potop-Butucaru, Cross hyperledger fabric transactions, in: CRYBLOCK 2020 - Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Part of MobiCom 2020, Association for Computing Machinery, 2020, pp. 35–40. doi:10.1145/3410699.3413796.
URL https://dl.acm.org/doi/10.1145/3410699.3413796

[132] D. Zhao, T. Li, Distributed Cross-Blockchain Transactions, arXiv (2020).

[133] X. Wang, O. T. Tawose, F. Yan, D. Zhao, Distributed Nonblocking Commit Protocols for Many-Party Cross-Blockchain Transactions (jan 2020). arXiv:2001.01174.
URL http://arxiv.org/abs/2001.01174

[134] X. Xiao, Z. Yu, K. Xie, S. Guo, A. Xiong, Y. Yan, A Multi-blockchain Architecture Supporting Cross-Blockchain Communication, in: Communications in Computer and Information Science, Vol. 1253 CCIS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 592–603. doi:10.1007/978-981-15-8086-4{_}56.
URL https://link.springer.com/chapter/10.1007/978-981-15-8086-4_56

[135] M. Qi, Z. Wang, D. Liu, Y. Xiang, B. Huang, F. Zhou, 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), Vol. 12404 LNCS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 154–168. doi:10.1007/978-3-030-59638-5{_}11.
URL https://link.springer.com/chapter/10.1007/978-3-030-59638-5_11

[136] Clearmatics, Ion Interoperability Framework v2 (2018).
URL https://github.com/clearmatics/ion

[137] Digital Asset, Canton : A Private , Scalable , and Composable Smart Contract Platform (2020) 1–15.

[138] Y. Pang, A New Consensus Protocol for Blockchain Interoperability Architecture, IEEE Access 8 (2020) 153719–153730. doi:10.1109/ACCESS.2020.3017549.

[139] E. Scheid, B. Rodrigues, B. Stiller, Toward a policy-based blockchain agnostic framework, 16th IFIP/IEEE International Symposium on Integrated Network Management (2019).

[140] M. Westerkamp, Verifiable Smart Contract Portability, ICBC 2019 - IEEE International Conference on Blockchain and Cryptocurrency (2019) 413–421.
URL http://arxiv.org/abs/1902.03868

[141] V. Buterin, R3 Report - Chain Interoperability, Tech. rep., R3 Corda (2016).
URL https://www.r3.com/wp-content/uploads/2017/06/chain_interoperability_r3.pdf

[142] C. Allen, The path to self-sovereign identity, Life with Alacrity (2016).

[143] G. Kondova, J. Erbguth, Self-sovereign identity on public blockchains and the GDPR, Proceedings of the ACM Symposium on Applied Computing (2020) 342–345doi:10.1145/3341105.3374066.

[144] N. Sakimura, NRI, J. Bradley, Ping Identity, M. Jones, Microsoft, B. Medeiros, Google, C. Mortimore, Salesforce, OpenID Connect Core 1.0 incorporating errata set 1 (2014).
URL https://openid.net/specs/openid-connect-core-1_0.html

[145] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, K. Klingenstein, Federated security: The Shibboleth approach, Educause Quarterly 27 (4) (2004) 12–17.

[146] R. Oppliger, Microsoft .NET Passport and identity management, Information Security Technical Report 9 (1) (2004) 26–34.

[147] C. Schläger, M. Sojer, B. Muschall, G. Pernul, Attribute-based authentication and authorisation infrastructures for e-commerce providers, in: Lecture Notes in Computer Science, 2006. doi:10.1007/11823865{_}14.

[148] M. Sporny, D. Longley, D. Chadwick, Verifiable Credentials Data Model 1.0: Expressing verifiable information on the Web (W3C Recommendation) (2020).
URL https://w3c.github.io/vc-data-model/

[149] R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, A Survey on Blockchain Interoperability: Past, Present, and Future Trends, arXiv (2020).
URL http://arxiv.org/abs/2005.14282

[150] H. Martins, S. Guerreiro, Access Control Challenges in Enterprise Ecosystems: Blockchain-Based Technologies as an Opportunity for Enhanced Access Control, Vol. i, 2018.

[151] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, M. Sabadello, J. Holt, Decentralized Identifiers (DIDs) v1.0: Core architecture, data model, and representations (W3C Working Draft) (2020).
URL https://w3c.github.io/did-core/

[152] M. Sporny, D. Longley, D. Chadwick, Verifiable Credentials Data Model 1.0 (2020).
URL https://www.w3.org/TR/vc-data-model/

[153] D. Ferraiolo, R. Kuhn, Role-Based Access Control, in: In 15th NIST-NCSC National Computer Security Conference, 1992, pp. 554–563.

[154] V. Hu, D. Ferraiolo, R. Chandramouli, R. Kuhn, Attribute-Based Access Control, Artech House, 2017.

[155] R. S. Sandhu, P. Samarati, Access Control: Principles and Practice, IEEE Communications Magazine 32 (9) (1994) 40–48. doi:10.1109/35.312842.

[156] R. Erik, OASIS eXtensible Access Control Markup Language (XACML) Version 3.0, Tech. rep. (2013).

[157] D. Di Francesco Maesa, P. Mori, L. Ricci, A blockchain based approach for the definition of auditable access control systems, Computers & Security 84 (2019) 93–119.

[158] L. Florio, K. Wierenga, Eduroam, providing mobility for roaming users, in: Proceedings of the EUNIS 2005 Conference, Manchester, 2005.

[159] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, Guide to attribute based access control ({ABAC}) definition and considerations, NIST Special Publication (2014). doi:10.6028/NIST.SP.800-162.

[160] H. L'Amrani, B. E. Berroukech, Y. El Bouzekri El Idrissi, R. Ajhoun, Toward interoperability approach between federated systems, in: ACM International Conference Proceeding Series, 2017. doi:10.1145/3090354.3090391.

[161] P. A. Bernstein, V. Hadzilacos, N. Goodman, Concurrency control and recovery in database systems, Addison-Wesley, 1987.

[162] S. Guerreiro, Challenges of Meta Access Control Model Enforcement to an Increased Interoperability, Vol. 43, 2018. doi:10.5860/choice.43-0002.

[163] S. Rouhani, R. Belchior, R. S. Cruz, R. Deters, Distributed Attribute-Based Access Control System Using a Permissioned Blockchain, arXiv pre-prints (2020).
URL http://arxiv.org/abs/2006.04384

[164] Hyperledger Contributors, Hyperledger Indy (2020).
URL https://www.hyperledger.org/use/hyperledger-indy

[165] Hyperledger, Hyperledger Aries (2020).
URL https://www.hyperledger.org/use/aries

[166] M. Hargreaves, T. Hardjono, R. Belchior, Open Digital Asset Protocol draft 02, Internet-Draft draft-hargreaves-odap-02, Internet Engineering Task Force (2021).
URL https://datatracker.ietf.org/doc/html/draft-hargreaves-odap-02

[167] T. Hardjono, A. Lipton, A. Pentland, Towards an Interoperability Architecture Blockchain Autonomous Systems, IEEE Transactions on Engineering Management 67 (4) (2019) 1298–1309.
URL doi:10.1109/TEM.2019.2920154

[168] R. Belchior, M. Correia, T. Hardjono, DLT Gateway Crash Recovery Mechanism draft 02, Internet-Draft draft-belchior-gateway-recovery-02, Internet Engineering Task Force (2021).
URL https://datatracker.ietf.org/doc/html/draft-belchior-gateway-recovery-02

[169] A. Sardon, T. Hardjono, Benedikt Schuppli, Asset Profile Definitions for DLT Interoperability (draft-sardon-blockchain-interop-asset-profile-00), Tech. rep. (2021).
URL https://datatracker.ietf.org/doc/draft-sardon-blockchain-interop-asset-profil

[170] J. S. Waterman, The Promissory Note as a Substitute for Money, Minnesota Law Review 14 (1929) 313.

[171] FQX, eNI™ Infrastructure - fqx.ch - Electronic Negotiable Instruments - FQX (2020).
URL https://fqx.ch/

[172] Transatlantic Shipment of Metals Financed via FQX eNote — Treasury Management International.
URL https://treasury-management.com/news/transatlantic-shipment-of-metals-finance

[173] T. Hardjono, N. Smith, Towards an Attestation Architecture for Blockchain Networks (to appear), World Wide Web Journal – Special Issue on Emerging Blockchain Applications and Technology (2021).
URL https://arxiv.org/abs/2005.04293

[174] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, B. Sniffen, Principles of remote attestation, International Journal of Information Security 10 (2) (2011) 63–81. doi:10.1007/s10207-011-0124-7.
URL https://link.springer.com/article/10.1007/s10207-011-0124-7

[175] H. Montgomery, H. Borne-Pons, J. Hamilton, M. Bowman, P. Somogyvari, S. Fujimoto, T. Takeuchi, T. Kuhrt, R. Belchior, Hyperledger Cactus Whitepaper (2020).
URL https://github.com/hyperledger/cactus/blob/master/docs/whitepaper/whitepaper.md

[176] Final: OpenID Connect Core 1.0 incorporating errata set 1.
URL https://openid.net/specs/openid-connect-core-1_0.html

[177] E. Rescorla, RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3 (2014).
URL https://tools.ietf.org/html/rfc8446

[178] S. Rouhani, R. Belchior, R. S. Cruz, R. Deters, Distributed attribute-based access control system using permissioned blockchain, World Wide Web (2021). doi:10.1007/s11280-021-00874-7.
URL https://doi.org/10.1007/s11280-021-00874-7

[179] S. Ghaemi, S. Rouhani, R. Belchior, R. S. Cruz, H. Khazaei, P. Musilek, A Pub-Sub Architecture to Promote Blockchain Interoperability, Future Generation Computer Systems (jan 2021). arXiv:2101.12331.
URL http://arxiv.org/abs/2101.12331