TÉCNICO LISBOA

An Enterprise Architecture Approach to Semantic Blockchain Interoperability

Sebastião Cristo Albernaz de Sotto-Mayor

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor(s): Prof. Miguel Nuno Dias Alves Pupo Correia
Prof. André Ferreira Ferrão e Couto e Vasconcelos

Examination Committee

Chairperson: Prof. Daniel Simões Lopes
Supervisor: Prof. André Ferreira Ferrão e Couto e Vasconcelos
Member of the Committee: Prof. Sérgio Luís Proença Duarte Guerreiro

November 2023

Dedicated to my family and friends

Acknowledgments

I want to thank my supervisors, Professors Miguel Correia, Andre Vasconcelos, and Rafael Belchior, for their unwavering guidance and commitment along this endeavor.

I am grateful to my parents and family for their enduring support throughout my academic journey, urging me to persevere.

Lastly, I thank my close friends and colleagues who provided invaluable advice and feedback during the development of this project.

Resumo

A tecnologia blockchain revolucionou o armazenamento e acesso de dados de forma descentralizada. No entanto, a falta de interoperabilidade entre estes sistemas é um desafio contínuo que impede a sua adoção em larga escala. Este documento propõe uma solução de duas partes composta por atividades que visam melhorar a interoperabilidade semântica entre sistemas blockchain homogéneos e heterogéneos. A primeira parte são as atividades de design que consistem na construção de um modelo Archimate, na extração da sua ontologia em RDF e na avaliação da sua retidão usando um raciocinador semântico. A segunda parte são as atividades de tempo de execução que envolvem o uso da ontologia resultante numa aplicação de gestão de cadeias de abastecimento para validar transações entre participantes numa rede de sistemas. Os resultados da avaliação são promissores, e demonstram que uma ontologia partilhada pode suportar uma abordagem transparente e precisa na validação de transações. Assim, este trabalho representa um passo significativo para provar que as tecnologias de registro distribuído podem beneficiar de técnicas de arquitetura empresarial para melhorar a sua interoperabilidade.

Palavras-chave: Arquitetura Empresarial, Interoperabilidade Blockchain, Ontologia

Abstract

Blockchain technology has revolutionized the way data is stored and accessed in a decentralized manner. However, the lack of interoperability between such systems is an ongoing challenge hindering their wider adoption. This document proposes a two-part solution composed of activities that aim to enhance semantic interoperability between homogeneous and heterogeneous blockchain systems. The first part are the design-time activities that consist of constructing an Archimate model, extracting its Resource Description Framework (RDF) ontology, and assessing its correctness utilizing a semantic reasoner. The second part are the runtime activities that involve leveraging the resulting ontology in a supply chain management application to validate transactions among participants in a network of systems. The evaluation results are promising, demonstrating that a shared ontology can support a transparent and accurate transaction validation approach. Thus, this work is a significant step in proving that distributed ledger technologies can benefit from enterprise architecture techniques to improve their interoperability.

Keywords: Enterprise Architecture, Blockchain Interoperability, Ontology

Contents

Acknowledgments v
Resumo vii
Abstract ix
List of Tables xv
List of Figures xvii
Nomenclature xix
Glossary xxi

1 Introduction **1**
1.1 Topic Overview 1
1.2 Motivation 2
1.3 Proposed Solution 2
1.4 Contributions 3
1.5 Thesis Outline 3

2 Related Work **5**
2.1 Blockchain ecosystems 6
2.1.1 Polkadot 6
2.1.2 Cosmos 7
2.1.3 Avalanche 7
2.1.4 Ark 8
2.1.5 MultiChain 8
2.1.6 Aion 9
2.1.7 Hyperservice 10
2.1.8 Wanchain 10
2.2 SDKs 11
2.2.1 Rosetta 11
2.2.2 Komodo 11

2.2.3 Hyperledger Cacti . . . 12
2.2.4 Quorum . . . 12
2.2.5 Blocknet . . . 13
2.2.6 Fusion . . . 14
2.3 Protocols . . . 14
2.3.1 Hephaestus . . . 14
2.3.2 Gas-Efficient Superlight Bitcoin Client . . . 15
2.3.3 XCLAIM . . . 15
2.3.4 AMHL . . . 15
2.3.5 Hermes . . . 16
2.3.6 Layer-One.X . . . 16
2.3.7 DeXTT . . . 17
2.3.8 Zendoo . . . 17
2.3.9 Quant Overledger . . . 18
2.3.10 Decentralized Cross-Network Identity Management for BI . . . 18
2.3.11 Cross-chain Deals and Adversarial Commerce . . . 18
2.3.12 Smart Contracts on the Move . . . 19
2.4 Blockchain Interoperability Observations . . . 19
2.4.1 Paradigms . . . 21
2.4.2 Trends . . . 21
2.4.3 Barriers . . . 22

3 Background **23**
3.1 Hyperledger Cacti . . . 23
3.2 Supply Chain Use Case . . . 24

4 Ontologically-Guided Blockchain Interoperability **27**
4.1 Design-Time Activities . . . 27
4.1.1 Enterprise Architecture Model Design . . . 27
4.1.2 Ontology Extraction and Assessment . . . 28
4.2 Runtime Activities . . . 31
4.2.1 Entity Mapping and Conversion . . . 31
4.2.2 Transaction Validation . . . 31

5 Evaluation **35**
5.1 Experimental Setup . . . 35

5.2 Accuracy Analysis 36
5.3 Performance Analysis 37
5.4 Result Discussion 38
5.5 Considerations 39

6 Conclusion **41**

Bibliography **43**

List of Tables

2.1 Related work system categorization and comparison 20

5.1 Experiment results 37

List of Figures

3.1 UML sequence diagram of an architecture supported by Hyperledger Cacti . . . 24
3.2 Supply chain use case BPMN diagram 26

4.1 Supply chain blockchain interoperability EA model in Archimate 29

5.1 Overhead of proposed solution 38

Nomenclature

Subscripts

$\|$	Parallel execution
#	Number sign
ms	Millisecond
Tx	Transaction

Glossary

API Application Programming Interface. 24

BA Blockchain Abstractions. 5

BFT Byzantine Fault Tolerance. 7

BI Blockchain Interoperability. 5

BLP Business Logic Plugin. 23

BPMN Business Process Model and Notation. 25

BoB Blockchain of Blockchains. 5

DAG Directed Acyclic Graph. 17

DLT Distributed Ledger Technology. 1

DPoS Delegated Proof of Stake. 8

DPoW Delegated Proof of Work. 11

EA Enterprise Architecture. 2

EIRA European Interoperability Reference Architecture. 27

HTLC Hashed time-lock contract. 5

ICT Information and Communication Technology. 21

IoV Internet of Values. 14

MVC-B Model-View-Controller-Blockchain. 12

NPoS Nominated Proof of Stake. 6

POC Proof of Concept. 2

PoI Proof of Intelligence. 9

PoP Proof of Participation. 16

PoS Proof of Stake. 7

REST Representational State Transfer. 24

SDK Software Development Kit. 24

S&R Sidechain & Relay. 5

Chapter 1

Introduction

This chapter provides a comprehensive overview of the research topic in Section 1.1, outlining the essential context for the study. In Section 1.2, it delves into the motivation behind the work, elucidating the specific technological gaps that prompted this research. The chapter also defines the proposed solution in Section 1.3, detailing how the study aims to achieve its objectives. Additionally, it states its contributions in Section 1.4 and offers a structured outline in Section 1.5, providing a roadmap of the subsequent chapters.

1.1 Topic Overview

Distributed ledger technology (**DLT**) refers to networks of computers that store databases in a distributed way, distinguishing them from traditional databases confined to a single server. This decentralized structure enhances security by making data tampering more challenging for malicious entities. Blockchains store data in a chain of blocks and are a specific type of DLT. Each block contains multiple transaction records, and once appended to the chain, a block becomes immutable. This mechanism establishes a permanent and transparent ledger of all transactions conducted in a network. In addition to being used to record and track monetary transactions, blockchain technology can potentially support applications in several other domains, including supply chain management, voting systems, and identity verification [Pec17].

Private or permissioned blockchains are distributed ledger systems where access to the network and validation of transactions is limited to a select group of entities. These blockchains provide enterprises with several advantages in comparison with permissionless blockchains. Firstly, data confidentiality is enhanced since sensitive information is shared only among trusted participants. Secondly, they are highly scalable due to reduced network congestion and faster transaction processing since the number of participants is reduced compared to public blockchains.

Additionally, private blockchains offer increased control and governance, enabling businesses to set rules and consensus mechanisms that align with their specific requirements. Furthermore, their usage enhances compliance, which makes them a valuable asset for various use cases in industries with high regulation [Und16].

Interoperability, which refers to the ability of different systems and components to work together efficiently, is a quality attribute that has four layers: the technical (further subdivided into functional and structural), the semantic (including interpreting information format and its meaning), the organizational, and the legal [KC19]. Regarding specifically the semantic layer of interoperability, on the one hand information format pertains to the structure and representation of data exchanged between systems. This includes data types, encoding methods, and communication protocols. On the other hand, understanding the meaning embedded in the information involves interpreting the data context, semantics, and relationships, ensuring that the exchanged information is correctly understood by the systems consuming it.

1.2 Motivation

Over the last few years, there have been many attempts to introduce a solution that will mitigate the interoperability issues of blockchain systems [BVGC21]. However, few solutions focus on the semantic layer. While each layer has its significance, the semantic layer will play a key role in increasing interchain communication efficiency [HLP19]. By building systems on a common ground of understanding, actions conducted among systems with a high volume of cross-chain transactions become faster, as there is less friction to translate concepts from one blockchain to the other.

1.3 Proposed Solution

The objective of this work is to explore enterprise architecture (**EA**) mechanisms to produce an ontology [ACB+13] that individual systems can leverage to build plugins, which are small and flexible components executing their business logic on top of a base architecture that includes the necessary elements to achieve interoperability. EA offers standardizable methods in the form of models that can facilitate the intercommunication of distributed ledger technology systems in the semantic layer by reducing ambiguity in valuable processes such as interchain transactions.

A proof-of-concept (**POC**) presents the practical aspects of this approach. Its scope is a network of permissioned blockchains owned by organizations with interoperation needs with fellow members of a supply chain consortium. The decision to restrict the POC to private

blockchains comes from the fact that systems involving public blockchains operate with some inherently different concerns and requirements as mentioned above. Therefore, designing an EA model targeting both would complicate the solution prematurely.

The results drawn from the conducted experiments are encouraging, showing that the integrity of the transaction validation process is enhanced when following the proposed approach.

1.4 Contributions

This thesis introduces an EA model that supports semantic interoperability between blockchain systems and implements a POC that will leverage the ontology extracted from it. By proving the applicability and effectiveness of the proposed solution, the aim is to show that EA techniques can be valuable for solving interoperability issues in blockchain systems.

1.5 Thesis Outline

This document is structured as follows. Chapter 2 presents the state of the art and Chapter 3 gives an overview of the architecture and use case that support the POC. Next, Chapter 4 describes the proposed solution in detail, and Chapter 5 provides the evaluation strategy and its outcome. Last, the conclusions drawn from this study are discussed in Chapter 6, including the project achievements and future work.

Chapter 2

Related Work

This chapter presents previous work that addressed the topic of blockchain interoperability (**BI**) and is therefore relevant to this thesis. Section 2.1, starts by providing an overview of existing ecosystems focused on interoperability. Next, Section 2.2 describes SDKs that enable interchain communication and Section 2.3 includes protocols that aim to solve specific issues in this area. Finally, Section 2.4 gives insight into the lessons learned from this endeavor, which proved significant when developing the proposed solution.

It is worth mentioning that there are currently no solutions tackling the enhancement of semantic BI other than the one presented in this thesis. Still, there has been an attempt to introduce an EA reference model specific for blockchain systems, although with no demonstrated practical use so far [ERH19]. On the other hand, transaction validation which is also a focal part of the work presented in this thesis has been explored by various tools, namely Bungee [BTP+22]. Furthermore, ontologies have been extensively utilized for the semantic integration of supply chain processes [Pet20].

Before delving into the concrete details of each technology, a description of the types of BI approaches is necessary. The three main types are sidechain & relay (**S&R**), notary scheme, and hashed time-lock contract (**HTLC**). In the first technique, one blockchain (mainchain) considers another blockchain as an extension of itself (the sidechain). The second approach involves an entity (notary) that monitors multiple chains, triggering transactions in a chain upon an event taking place on another chain. In the HTLC mechanism, a trader commits to make a transaction by providing cryptographic proof before a timeout occurs. Furthermore, there are two other types that employ techniques from the ones defined previously. Those are the blockchain of blockchains (**BoB**) that can be seen as a generalization of S&R, and blockchain abstractions (**BA**) that are solutions that seek to hide the complexity of the implementation of each system [BVGC21].

The following systems are presented with no particular order of significance, as the aim is to present their characteristics and not to classify them. For each, there is a brief description of its architecture, its strengths and limitations, and its unique contributions to the field. A detailed comparison can be found in Table 2.1.

2.1 Blockchain ecosystems

A blockchain ecosystem is a network of various stakeholders involved in the creation, maintenance, and evolution of a blockchain-based system or network. This ecosystem typically includes a mix of participants such as users, developers, miners, businesses, and governments, all of whom have a vested interest in the success and growth of the blockchain. The ecosystem is driven by the rules and incentives built into the blockchain network, and the interactions and transactions between these stakeholders help to secure and validate the blockchain.

2.1.1 Polkadot

Polkadot [Woo16] is a scalable, heterogeneous multi-chain system with the potential to be backwards compatible to certain, preexisting blockchain networks.

The main participants in the Polkadot network are the Validators, the Nominators that approve Validators, the Collators that provide block candidates, and the Fishermen that monitor Validators and report illegal behavior with a fee incentive. Designers of Polkadot modeled its Relay Chain as a state machine, which impossibilitates double spending and ensures transaction validity. They are responsible for block production and validity, availability, and cross-chain message passing. Furthermore, governance is dealt with by each Parachain independently, and all of them benefit from baseline security. Parachain is the name given to the sidechains of the network. The consensus approach of the network is the Nominated Proof of Stake (**NPoS**), which is decentralized and secure. It introduces novel algorithms such as BABE and GRANDPA with probabilistic and deterministic finality correspondingly. Lastly, it leverages Bridge Chains for communication with external blockchains.

Polkadot's strengths are that it is more efficient compared to other platforms and consumes less power due to the novel NPoS consensus algorithm, its scalability because of the decoupling of its consensus and state-transition modules, and its isolatability because of the multiple application needs served under a single platform. Also, it offers trust-free access to its data from open and closed networks. Its limitation is the uncertainty of backward compatibility with transitioning blockchains.

2.1.2 Cosmos

Cosmos [KB19] is a network connecting many independent blockchains. A high-performance, consistent, secure practical Byzantine Fault-Tolerant (**BFT**)-like consensus algorithm is employed, where strict fork accountability guarantees hold over the behavior of malicious actors.

The participants of the Cosmos network are the Validators (similar to the previous ecosystem). The names given to its mainchain and sidechain are Hub and Zones, correspondingly. The former is responsible for double-spending and censorship mitigation and ensuring transaction validity and baseline security to all Zones, while the latter deal with governance independently. The network reaches consensus by Proof of Stake (**PoS**). Finally, its inter-blockchain communication mechanism ensures safe interoperability between Zones, achieved by leveraging transaction types, and communication with external blockchains happens from Bridge Zones.

Cosmos's strengths are its high performance, strict fork accountability because of its punishing policy for processes that cause a fork, and low power consumption resulting from the PoS approach. One of its limitations is that its Validator number has an eventual limit because of the high complexity involved in communication. Also, masking of distributed attack origins is possible because of network partitioning practices.

2.1.3 Avalanche

Avalanche [SLBS20] is a blockchain platform for creating, transferring, and trading digital assets while maintaining high performance, among other significant quality attributes.

Its architecture has the following set of properties. First, it supports a global network of interconnected devices, proving it is scalable. It is secure as it provides strong safety when the attacker is below a certain threshold and continues to do so, albeit without liveness when the attacker exceeds it. Also, it is decentralized because its design avoids divisions between classes of users with different interests and enforces no centralized control whatsoever. It is governable and democratic, as any token holder has a vote in selecting key financial parameters and choosing how the system evolves. Lastly, it is interoperable and flexible, resulting from the design that makes it easy to port existing blockchains and support multiple scripting languages and virtual machines. The sidechains, or Subnetworks in Avalanche, leverage a dynamic set of validators working together to achieve consensus on the state of several blockchains. The network employs a consensus engine based on protocols that operate by repeated network sampling. Each node polls a small, constant-sized, randomly chosen group of neighbors and switches its proposal if a supermajority supports a different value. This process continues until nodes converge, which happens rapidly in normal circumstances.

Avalanche's strengths are that Snow protocols combine the best properties of classical consensus protocols with the best of Nakamoto consensus, thus providing scalability without high energy consumption. Also, it promotes fairness by employing an egalitarian distribution of minting where participants in the staking protocol get equitable rewards proportional to their stake. Furthermore, each participant in the protocol has a voice in influencing how the protocol evolves, made possible by its governance mechanism. Lastly, it supports high customizability, allowing instant plug-and-play with existing blockchains. Its limitation is the dependency of participants on the platform's native token.

2.1.4 Ark

Ark [ARK19] is a user-friendly platform that aims to increase user adoption of blockchain technology.

Its architecture features a consensus algorithm that is a variation of Delegated Proof-of-Stake (**DPoS**) where the voting weight of each wallet is split evenly between all voted delegates. Also, it offers a mechanism to bridge blockchains via a function built into ARK Core where any blockchain can send and receive trigger function notices and informational data through the primary network via custom-developed SmartBridges (Ark's sidechains) and Encoded Listeners. Furthermore, it offers several off-chain tools such as multi-account management, microtransactions, and local currencies.

Ark's strength is its code association with mature platforms that provides simplified future interaction with other blockchain systems using DPoS as their consensus. Some limitations are that platform participants become dependent on its native token and that it may become too coupled with supporting platforms (i.e., Lisk, Crypti and BitShares).

2.1.5 MultiChain

MultiChain [G$^+$15] is an off-the-shelf platform for creating and deploying private blockchains within or between organizations. It aims to overcome a key obstacle in adopting DLT in the institutional financial sector by providing the privacy and control required in an easy-to-use package.

Its architecture relies on a consensus mechanism of permissions-based mining with optional PoW restricted to specific entities.

MultiChain's strengths are the ease of merge of future bitcoin enhancements as it is a fork of Bitcoin Core, the official client for the bitcoin network with localized code changes, and the ability for fine-grained permissions with an option to control who can connect, transact, create

assets, streams, and blocks, so chains are as open or as closed as needed. Its limitation is that network consensus may become compromised because of the possibility of a permitted entity in a private blockchain going rogue.

2.1.6 Aion

Aion [ST+17] is a network built to support custom blockchain architectures while providing a trustless mechanism for cross-chain interoperability. At the root of this system is a dedicated public enterprise blockchain called Aion-1, which introduces a paradigm of security and crypto-economic incentives.

A set of goals defines Aion's architecture. Connecting chains and external services (e.g., oracles and DBs) and providing accountability through network contiguity and decentralization, providing the necessary infrastructure to develop high-performance and interoperable applications, and creating a maintainable network through a robust and sustainable economic model. Its consensus mechanism includes two variations of the BFT protocol combined with Proof of Intelligence (**PoI**) which requires participants to train a predefined neural network so that it will output equivalent results to the proposed ground truth (e.g., the hash of the current block given the hashes of previous N blocks as input). The parameters of the trained neural network serve as proof that computation took place and is easy to verify by inputting the parameter and confirming the result. Specifically, the variations mentioned above are bridge consensus and connecting network consensus. The former is a lightweight variation to reach an agreement on the bridge fast, and the latter focuses on providing stability at scale. There are two types of networks in the platform. Those are the connecting networks that facilitate interchain communication and transactions between multiple private or public blockchain networks and the participating networks that can be purpose-specific blockchains, private networks, or consortium blockchains representing collections of entities. These networks then leverage two mechanisms to accomplish their tasks. Those are the interchain transaction which is a trust-free message between blockchain networks, a critical infrastructure component powering interchain communication, and the bridge, a communication protocol composed of its own distinct set of validators that assures translation of protocols and accountability between networks.

Aion's strength is the introduction of the technique of varied consensus according to contextual needs, where one can have a choice about the tradeoff between security and performance. Its limitation is the non-triviality of existing BI, for which one requires additional assumptions and compromises.

2.1.7 Hyperservice

Hyperservice [LXS+19] is a platform that delivers interoperability and programmability across heterogeneous blockchains. Powered by two key designs; a developer-facing programming framework that allows developers to build cross-chain applications in a unified programming model; and a secure blockchain-facing cryptography protocol that provably realizes those applications on blockchains.

Its architecture's main components are the dApp clients, which are gateways for dApps to interact with the platform, verifiable execution systems, which are blockchain drivers that compile the high-level dApp programs given by the dApp clients into blockchain-executable transactions, the Network Status Blockchain, a blockchain of blockchains that provides an objective and unified view of the execution status of dApps, and Insurance Smart Contracts which arbitrate the correctness or violation of dApp executions. Other key features of the network are its Universal State Model, a blockchain-neutral and extensible model for describing state transitions across different chains, and the Universal Inter-Blockchain Protocol, a cryptography protocol between VESes and dApp clients to execute HSLs securely.

Hyperservice's strength is that it is three orders of magnitude faster than the speed of Ethereum mainnet. Its limitation is that its steep learning curve may deter large-scale adoption of the platform.

2.1.8 Wanchain

Wanchain [Wan22] is a platform that connects and exchanges value between different blockchain ledgers in a distributed manner. It uses the latest cryptographic theories to build a nonproprietary cross-chain protocol and a distributed ledger that records both cross-chain and intra-chain transactions. Any blockchain network, whether public, private, or consortium chain, can integrate to establish connections between different ledgers and perform low-cost inter-ledger asset transfers. This ledger supports not only smart contracts but also token exchange privacy protection.

Its main design objectives are cross-chain asset transfers to connect existing major digital currency networks and integrate consortium chains, ensuring the security and stability of transactions and transaction privacy protection which will be customizable by trading parties with anonymization capabilities in transfers and exchanges of digital assets and functional extensibility. Wanchain's consensus mechanism is PoS with verification nodes, and the components that make up the system are the Vouchers which are cross-chain transaction proof nodes, the Storemen that are locked account management nodes and the Validators that are general verification

nodes.

Wanchain's strength is that it provides high-security guarantees by employing state-of-the-art cryptography (e.g., multi-party computation, threshold secret-sharing, ring signatures, and one-time account usage). Its limitation is that participants of the ecosystem become dependent on its native token.

2.2 SDKs

Besides complete ecosystem solutions, there also exist SDKs that facilitate BI without providing a unifying platform, unlike the systems presented in the previous section, but by giving the means for separate networks to become interoperable among themselves.

2.2.1 Rosetta

Rosetta [Cha] is a platform that aims to make blockchain integration simpler, faster, and more reliable than using a native integration.

Its architecture offers a plethora of APIs for implementing sidechains on top of a Relay Chain. Those APIs fall into two main categories; the Data API and the Construction API. The former offers sidechain inter-communication, and the latter facilitates the incorporation of features to sidechains by providing base functionality.

Rosetta's strengths are its steep learning curve because of its language-agnostic APIs and the single notion of transaction-related entities and standard data format (significantly relevant to the current work efforts).

2.2.2 Komodo

Komodo [Lee18] is a decentralized exchange that removes all forms of intermediaries, vouchers, and escrow services. It introduces a form of security that is as strong as the Bitcoin [Nak08] network yet does not require the expensive cost.

Its components are Smart Chains, the Antara framework, which enables the development of implementation-specific features, and AtomicDEX, a decentralized exchange. The consensus mechanism used is Delayed Proof of Work (**DPoW**).

Komodo's strengths are that it is bundled with ready-to-use white-label solutions thanks to the Antara Integration Layer, the fact that atomic cross-chain swaps [Her18] are possible among the vast majority of blockchains, and that the sidechains are entirely independent of the platform, other sidechains, and coins. Furthermore, Antara modules are native, Turing complete, and do

not require gas fees. Its limitation is the performance issues caused by imminent duplication in the Antara Module Library as the platform scales.

2.2.3 Hyperledger Cacti

Hyperledger Cacti [VGJ+16] is a system that enables business-to-business and business-to-customer transactions.

Its architecture has a set of requirements. Identity transparency and auditability by being possible for a party to prove its identity and ownership of an asset and for regulators to investigate transaction records. Privacy preservation and confidentiality by providing mechanisms to conceal the identity, transaction patterns, and contract terms from unauthorized third parties. Modularity by utilizing pluggable algorithms for users to select during deployment. Compliance by having code that provides the capability to restrict the functionality of the execution environment and the degree of computing flexibility according to regulations. Lastly, performance and scalability by completing functions in user-acceptable timeframes and handling significant validator expansion without degradation. Hyperledger provides three kinds of services. Namely, membership, blockchain, and chaincode services. The first focuses on registration, identity management, and auditability functions. The second service includes the P2P protocol, distributed ledger, storage, and consensus manager. The last service offers secure containers and registries. It also provides APIs for multiple actions and follows the Model-View-Controller-Blockchain (**MVC-B**) pattern, a variation of the popular MVC pattern that enhances controller logic with chaincode and the data model with transactions on the blockchain.

Hyperledger's strength is that it allows for compliance with regulations while supporting the varied requirements that arise when competing businesses work together on the same network. Its limitation is the reliance on community contribution for crucial system components such as consensus algorithm module implementations.

2.2.4 Quorum

Quorum [Hou18] is a permissioned blockchain based on the official Go implementation of the Ethereum [B+14] protocol.

Its architecture has four main components. Those are the transaction manager that allows access to encrypted transaction data and manages local data storage and communication, the crypto enclave, responsible for private key management and the encryption and decryption of sensitive transaction data, the QuorumChain, which is a BFT-hardened consensus mechanism that utilizes core Ethereum features to verify and propagate votes through the network and

lastly the network manager that provides access control, enabling a permissioned network to be created.

Quorum's strength is the inheritance of a production-hardened network's maturity (Go-Ethereum). Also, it helps unite the public and enterprise development communities on the same protocol (much like what the proposed solution intends to achieve). Its limitations are that it is not technology independent as it works on top of the Ethereum blockchain and that private transactions cannot change the public state because doing so would break the block validation/consensus algorithm.

2.2.5 Blocknet

Blocknet [CM18] is a foundational infrastructure for the token ecosystem that provides true peer-to-peer interoperability between nodes on different blockchains. It achieves this through an architectural and protocol-based approach.

Its main design objectives are interoperability by supporting most chain implementations and centralized entities to make traditional server-based services available, Decentralization so no one entity may be able to exercise control over other entities in the ecosystem, security for high determinacy of operation at a level comparable to aeronautical applications, trustless service delivery to not be necessary to trust a counterparty to act honestly throughout a transaction, integrability in a way that accesses to the token ecosystem shall not require modification of stock wallets or nodes, composability by being mindful of which services will always be consumed together, to avoid building a distributed monolith, monetizability that is a service must be intrinsically monetizable, else it should be bundled into a monetizable API to incentivize its hosting, and frugality by enabling mobile applications with a small footprint to consume and pay for inter-chain services without hosting a blockchain locally. The first of Blocknet's guiding principles is that inter-chain infrastructural services must run on the edges of both their network along with any service-delivery and consumption networks. Also, running components on the same machine achieves service decentralization when they support either the delivery or the consumption of a service. Lastly, inter-chain infrastructural services must limit their integration requirements and footprint where possible. Its core components are XBridge, an inter-chain network overlay, a blockchain router called XName, and XChat, a p2p data transport.

Blocknet's strength is its microservices-based architecture that enables ease of future extensibility. Its limitation is the possible sidelining of unmonetizable services resulting from its design philosophy.

2.2.6 Fusion

Fusion [Fus17] is an infrastructure that will connect various values by establishing a layer of control management on top of multiple tokens through a distributed management of their private keys and by providing ports both for central organizations and for external data sources and thus solve the problem of insufficiency in interoperability of the current Internet of Values (**IoV**).

Its architectural layers are the Hierarchical Hybrid Consensus Mechanism (HHCM) layer, the distributed control rights management layer, the cryptofinance-oriented smart contract layer, the cryptofinance Distributed Service (DSrv) layer, and finally, the cryptofinancial Dapp layer.

Fusion's strength is its clear architecture that enables an efficient, secure, and scalable solution. Its limitation is the reliance on PoW consensus, an outdated mechanism in the field of BI solutions.

2.3 Protocols

The last category of systems is called protocols in this document. These solutions introduce a novel approach to mitigate particular difficulties in this field but are not yet complete solutions with active users. Therefore, the real-world performance of these systems cannot yet be determined as there is no available data regarding their usage in that context.

2.3.1 Hephaestus

Hephaestus [BSP+22] is a framework for analyzing and evaluating the performance of cross-chain transactions, which are transactions that involve the transfer of assets or information between two different blockchain networks. The framework consists of a model of cross-chain transactions that includes various factors that can affect their performance, such as the number and type of nodes involved, the size and complexity of the transactions, and the network conditions.

The authors of the paper present several performance evaluation techniques that can be used to analyze the behavior of cross-chain transactions under different scenarios and conditions. They also describe several real-world case studies in which the Hephaestus framework was applied to evaluate the performance of cross-chain transactions in existing systems, such as the Ethereum and Bitcoin networks.

Overall, the Hephaestus framework provides a useful tool for understanding and optimizing the performance of cross-chain transactions, which are an important aspect of the broader field of blockchain technology.

2.3.2 Gas-Efficient Superlight Bitcoin Client

This solution [DKKZ20] is an on-chain decentralized client implemented in Solidity that securely verifies cross-chain events. It establishes a trustless and efficient solution to the interoperability problem.

Its architecture leverages two key techniques. The first is the hash-and-resubmit pattern, which reduces gas consumption by utilizing the call data space of the Ethereum blockchain to eliminate high-cost storage operations. The second is the optimistic schema, which achieves significant performance improvement by replacing linear complexity verification of proofs with constant complexity verification.

This solution's strength is that it achieves gas cost reduction while maintaining high availability and consistency, while its limitation is that it is currently limited to transactions from Bitcoin to Ethereum.

2.3.3 XCLAIM

XCLAIM [ZHL+19] is a secure system to construct cryptocurrency-backed assets without trusted intermediaries.

Its architecture consists of a specific set of actors. The CBA Requester who locks b on B to request asset i be backed by b on I, the CBA Sender who owns the asset I that is backed by b and transfers ownership to another user on I, and the CBA Receiver who receives and is assigned ownership over the asset I that is backed by b on I. Furthermore, there is the CBA Redeemer who destroys the asset i that is backed by b on I to request the corresponding amount of b on B, the CBA Backing Vault who is a non-trusted intermediary liable for fulfilling redeem requests of the asset I that is backed by b on B, and the Issuing Smart Contract which is a public smart contract responsible for managing the correct issuing and exchange of the asset i that is backed by b on I.

XCLAIM's strengths are that it requires only base-ledger functionality on the backing side, supporting all cryptocurrencies and that compared to HTLC atomic swaps, it is 95.7% faster and 65.4% cheaper for 1000 swaps. Its limitations are that the blockchain used to issue cryptocurrency-backed assets must support smart contracts and that, for the moment, it does not address some attack vectors.

2.3.4 AMHL

AMHL [MMSS+18] is the first interoperable and privacy-preserving cryptographic construction for multi-hop locks.

A set of properties drive its architecture. Firstly, it must preserve atomicity so every user in a path can release their left lock in case they release their right lock. Consistency is also significant, so no attacker releases the left clock before the right one. Finally, relationship anonymity so that each intermediate node does not learn any information about the set of users beyond its direct neighbors.

AMHL's strengths are primarily that it significantly reduces the communication and computation overhead required by multi-hop Hash Time-Lock Contracts. Also, the overhead per hop is fast when TEE is used instead of cryptographic operations for the multi-hop locks. Furthermore, it deploys with ease in all cryptocurrencies. Its limitation is that the blockchain needs to support the verification of a homomorphic one-way function in their scripting language which nevertheless is an issue that the usage of a scriptless-construction variation may overcome.

2.3.5 Hermes

Hermes [BVCH22] is a fault-tolerant middleware that leverages its main component, a new two-phase protocol based on the original Open Digital Asset Protocol (ODAP-2PC), to provide ACID properties for cross-blockchain transactions.

A list of properties defines its architecture. Firstly it should ensure atomicity so that transactions either commit on all underlying ledgers or entirely fail, consistency so that all gateways that decide on a transaction reach the same decision, either commit or abort, and isolation so that when a transaction is issued, all the underlying assets are locked. Additionally, it should provide durability so that once a transaction commits, it remains so regardless of any component crashes, auditability so that the involved parties may inspect any executed transaction, and finally, termination so that if a gateway proposes a transaction, it eventually commits or aborts.

Hermes's strengths are its flexible and modular architecture, resulting from the pluggability of its components. Also, asset transfers are atomic and fair, and no double spending can occur. Its limitation is that it assumes underlying ledgers where gateways operate are safe. The reason is that the ODAP-2PC protocol is crash fault-tolerant, so it does not tolerate Byzantine faults (i.e., gateways that behave arbitrarily).

2.3.6 Layer-One.X

Layer-One.X [CCA+21] is a decentralized ledger that utilizes the Proof of Participation (**PoP**) consensus mechanism. It offers two solutions in one, which are Layer-One.Scalability and Layer-One.Interoperability. The present document, focuses on the latter.

Its architecture follows a set of steps. Firstly, it provides identity management which allows for identifying the sender and receiver in the transaction and flash contract that allow for the code to be executed on the nodes by pooling a distributed set of a required resource. Additionally, it enables settlement using customized directed acyclic graphs (**DAG**s) that maintain null pointers by a checksum strategy that allows for minimal resources to be checked and nucleus scripting and asset locking that run through sharding and send the updated global state to the network when it changes.

Layer-One.X's strength is that it meshes interoperability and scalability in the same solution. Its imitation is that only a prototype exists at this stage, so there are no benchmarks available to judge its usage in a production environment.

2.3.7 DeXTT

DeXTT [BSF+19] is a cross-blockchain transfer protocol, which can transfer tokens on any number of blockchains simultaneously in a decentralized manner.

Its architecture leverages a novel technique called cross-blockchain balance consistency that requires eventual consistency between blockchains participating in the proposed protocol, using claim-first transactions.

DeXTT's strength is its reliance on a pan-blockchain token (PBT) and thus does not restrict users to a blockchain-specific token. Its limitation is that using too short validity periods leads to corrupted transfers, as eventual consistency among blockchains becomes compromised.

2.3.8 Zendoo

Zendoo [GKO20] is a solution that allows the creation and communication with sidechains of different types without knowing their internal structure.

Three components enable its architecture. These are the Mainchain Consensus Protocol (MCP), the Cross-Chain Transfer Protocol (CCTP), which facilitates forward and backward transfers, and the Sidechain Consensus Protocol (SCP).

Zendoo's strength is that it improves scalability by leveraging a novel technique to construct proofs of state transitions, called the recursive succinct non-interactive argument of knowledge composition. Its limitation is that it aims specifically at Bitcoin-like mainchains that maintain a parent-child relationship with their sidechains, or in other words, it is not general nor 100% decentralized.

2.3.9 Quant Overledger

Quant Overledger [VTPM18] is an infrastructure for designing, deploying, and executing multi-ledger decentralized applications.

Its architecture is universal and general-purpose. Firstly, it achieves interoperability by conducting operations across multiple blockchains simultaneously, and its communication layer is a presentation layer on top of blockchains that allows applications to run on them. Additionally, its communication method adopts a two-phase commit schema, and the speed is proportional to the latency of the involved chains and only requires a fixed set of transactions.

Quant Overledger's strengths are its purpose reach, interoperability capabilities, and communication layer technique, while its limitation is the vagueness of scalability and fault tolerance strategies.

2.3.10 Decentralized Cross-Network Identity Management for BI

This solution [GRG+21] aims to design a secure distributed identity management infrastructure and set of protocols linking permissioned networks and laying the basis for blockchain interoperation.

Its architecture has a specific set of building blocks. Firstly, it possesses the Decentralized Identifiers so that each network participant possesses a DID, decoupling its external identity from its network affiliation and Verifiable Credentials. It also has the Distributed Verifiable Data Registry and the Identity and Credential Messaging, a platform-neutral data-sharing protocol. More specifically, its architecture comprises the Distributed Identity Infrastructure, which includes the Interoperation Identity Networks, the Trust Anchors, and IIN artifacts such as the real-world DIDs, Membership and Memberlist VCs, and Revocation Registries. Additionally, there are the Network Identity Managers, which include the IIN Agent and ledger artifacts such as the Interoperation Network and Trust lists and the Foreign Network Identities.

This solution's strengths are its generality and flexibility, security, and ease of extensibility. Its limitations are the Trust Anchor that issues real-world DIDs to organizations, is a centralized component, and the coupling of network participant identities to their IIN identities, which nevertheless affects liveness but not safety.

2.3.11 Cross-chain Deals and Adversarial Commerce

This solution [HLS22] introduces the cross-chain deal, a computational abstraction for structuring complex distributed exchanges in an adversarial setting (a superset of cross-chain swaps).

Its architecture possesses three main properties, the first being about safety and the others

about liveness. Firstly, for every protocol execution, every compliant party ends up with an accepted payoff. Also, it does not escrow forever any asset belonging to a compliant party. Lastly, if all parties comply and are willing to agree to their proposed payoffs, then all transfers happen. The steps to achieve interoperability are straightforward. The process begins with the Clearing stage, proceeds to the Escrow, then Transfer and Validation stages, and ends with the Commit stage. The protocols enabling this mechanism are two. These are the Timelock, which is synchronous and timeout-based, and the CBC, which is semi-synchronous and vote-based.

This solution's strengths are its low gas costs in CBC protocol implementation and the time costs in both protocols that are constant or no more than linear. Its limitations are the high gas costs for the Commit and Abort stages in the Timelock protocol implementations with many participants that can become quadratic, the possible censorship in CBC protocol implementations, and the denial-of-service vulnerability in CBC protocol implementations with a small delta.

2.3.12 Smart Contracts on the Move

This solution [FBP20] introduces the Move operation, which allows programmable blockchains to interoperate in two steps. Explained briefly, the first step locks a smart contract in the source blockchain. A second step then recreates the smart contract in the target blockchain in a provably correct way.

Its architecture builds on top of three assumptions for participating chains. These are the support for smart contracts, usage of the same execution environment (e.g., Ethereum Virtual Machine), and the ability to prove state variables succinctly (e.g., Merkle trees).

This solution's strengths are its prevention of replay attacks suitable for PoW and PoS mechanisms, the possibility of state and asset transfers, and the developer-induced extensibility approach. Its limitation is the Solidity language lock-in.

2.4 Blockchain Interoperability Observations

Thorough research into the state-of-the-art and technical literature has led to the compilation of some empirical observations about the field of interoperability in blockchain systems. The following subsections categorize them into paradigms, trends, and barriers to achieving interoperability.

System Name	Category	Type	Consensus	Token	Smart Contracts
Polkadot	Ecosystem	BoB	NPoS	DOT	+
Cosmos	Ecosystem	BoB	PoS	ATOM	+
Avalanche	Ecosystem	S&R	Snow	AVAX	+
Ark	Ecosystem	BoB	DPoS	ARK	±
MultiChain	Ecosystem	S&R	PBFT	MULTI	+
Aion	Ecosystem	S&R	adaptive	AION	+
Hyperservice	Ecosystem	S&R	adaptive	-	-
Wanchain	Ecosystem	HLTC	PoS	WAN	+
Rosetta	SDK	S&R	N/A	N/A	+
Komodo	SDK	BoB	N/A	N/A	+
Hyperledger Cacti	SDK	S&R	N/A	N/A	-
Quorum	SDK	S&R	N/A	N/A	+
Blocknet	SDK	S&R	N/A	N/A	+
Fusion	SDK	HLTC	N/A	N/A	+
Hephaestus	Protocol	BA	N/A	N/A	N/A
GESBC	Protocol	BA	N/A	N/A	N/A
XCLAIM	Protocol	HLTC	N/A	N/A	N/A
AMHL	Protocol	HLTC	N/A	N/A	N/A
Hermes	Protocol	BA	N/A	N/A	N/A
Layer-One.X	Protocol	BA	N/A	N/A	N/A
DeXTT	Protocol	HLTC	N/A	N/A	N/A
Zendoo	Protocol	BA	N/A	N/A	N/A
Quant Overledger	Protocol	BoB	N/A	N/A	N/A
DCNIMfBI	Protocol	BA	N/A	N/A	N/A
CCDaAC	Protocol	BA	N/A	N/A	N/A
SCotM	Protocol	BA	N/A	N/A	N/A

Table 2.1: Related work system categorization and comparison

2.4.1 Paradigms

There are several approaches to designing interoperability solutions for blockchain systems. The most prominent ones are implementing interoperability as a cross-cutting component [Woo16], as a background layer [CZDK17], or by following an interoperability framework [KK19].

The first approach is to have integrated public service governance. Achieving this requires holistic management of interoperability activities across administrative levels and sectors, setting processes to select relevant standards and specifications, evaluating them, and monitoring their implementation and compliance. Also, using a structured, transparent, objective, and a common approach to assessing and selecting standards and specifications and consulting relevant catalogs of standards, specifications, and guidelines when procuring and developing information and communication technology (**ICT**) solutions is of high importance. Lastly, one should actively participate in standardization work according to needs to ensure that all requirements go through.

The second strategy has to do with interoperability governance. Achieving this requires instituting the necessary governance structure, that is, the decision-making process, requirements, change management, and recovery plans, and establishing interoperability in all layers, complemented by operational agreements and change management procedures.

For the third approach, one has to ask the 4W and H questions. Firstly, "what" data objects or assets does the solution need to handle, and "who" controls the cross-chain transaction process and thus accounts for trust establishment. The "where" refers to the source and target ledgers. The "when" aims to understand the processes executing at design time or runtime. Finally, the last question is "how" do the underlying DLTs implement the cross-chain transaction and the testing approach for said implementation.

2.4.2 Trends

Presently, there are unique conditions to implement a robust solution for interoperability. The technology is mature enough to deal with many different implementations, incompatible abstractions, and the heterogeneity of stacks. Also, experimentation creates opportunities for gradually adopting and expanding network boundaries [Pan21].

Consequently, researchers have shifted their focus to topics such as the ability to move an asset from chain A to chain B and the atomic exchange of assets from chain A to another chain B. Also interesting are the cases of asset encumbrance to lock and unlock an asset in chain A depending on an event of chain B. Another area is that of general cross-chain contracts to be able to use data from chain A in operations conducted in chain B [But16].

When all these techniques become refined, it will finally be possible to think about data silo removal for the seamless flow of value among participants and network scale and growth while preserving trust and security. Furthermore, orchestrating complex cross-network business functionality will become a reality, leading to a significant increase in market size, liquidity, and efficiency of blockchain applications.

2.4.3 Barriers

Despite promising breakthroughs from projects in this field, there are still some open issues to overcome. The gap between theory and practice, including the lack of standardization, and network discoverability, are among the most challenging. Additionally, one must not forget the ever-important issue of privacy and security that the extensive literature repeatedly mentions as a tradeoff of interoperability and the eventual need for complex governance tactics [AHI+21].

Especially concerning to our research is the topic of bidirectional communication across permissioned ledgers, that as described in [NSER20], is being hampered by the lack of semantic interoperability.

Chapter 3

Background

This chapter presents important knowledge concerning the proposed solution. Specifically, Section 3.1 provides an in-depth analysis of Hyperledger Cacti [MBPH+20], as it forms the basis of the architecture that supports business logic plugins (**BLP**s) used for the interoperation of supply chain-related entities. Additionally, Section 3.2 describes the use case that will demonstrate the capabilities of the proposed solution along the remainder of the thesis.

3.1 Hyperledger Cacti

Blockchain gateways are intermediary systems that facilitate interoperability between different blockchain networks. They act as bridges, enabling data transfer across disparate blockchain platforms, allowing them to exchange information and assets. One example of a middleware framework that adopts the concept of blockchain gateways is Hyperledger Cacti. Built on the Hermes model [BVCH22], Cacti is an open-source solution that provides a fault-tolerant middleware layer supporting BI.

By implementing the protocols and mechanisms introduced by Hermes, Cacti enables secure and reliable communication between diverse blockchain networks, opening up opportunities for cross-chain transactions and the execution of smart contracts, which are self-executing contracts with the terms of the agreement written in a programming language like Solidity [Dan17]. These contracts automatically execute and enforce themselves when predetermined conditions are met, without the need for intermediaries or third parties.

The main components of Cacti are the following:

- *Core Framework*: The core framework forms the foundation of Cacti. It provides the necessary infrastructure and interfaces for developing blockchain integration modules. It includes the main runtime, messaging protocols, and a plugin system for extensibility.

- *Connector*: Connectors are modules that enable communication and interaction with various blockchain platforms. Each connector is designed to interface with a specific blockchain platform, such as Avalanche [SLBS20], Cosmos [KB19], and Corda [Bro18], to name a few. Connectors allow Cacti to interact with different blockchains simultaneously.

- *Consortium Management*: Cacti includes a consortium management component, which facilitates the creation and management of consortium networks. It allows multiple blockchain platforms to collaborate and communicate securely within a shared network.

- *Plugin System*: Cacti employs a plugin system that allows developers to extend its functionality by creating custom connectors and modules. This extensibility enables integration with new blockchain platforms or the addition of custom features and functionalities.

- *APIs and SDKs*: Cacti provides a set of application programming interfaces (**API**s) and software development kits (**SDK**s) that simplify the integration process and allow developers to build applications on top of the framework. These APIs and SDKs offer standardized interfaces for accessing the functionality provided by Cacti.

Figure 3.1 depicts the high-level interactions of components utilizing Cacti. The client communicates with multiple BLPs via Representational State Transfer (**REST**) APIs. The BLPs then send the data to Cacti Core, which serves as the central component responsible for coordinating the exchange of data between the plugins and the connectors via connector APIs. Lastly, each connector is responsible for interfacing with the respective blockchain.

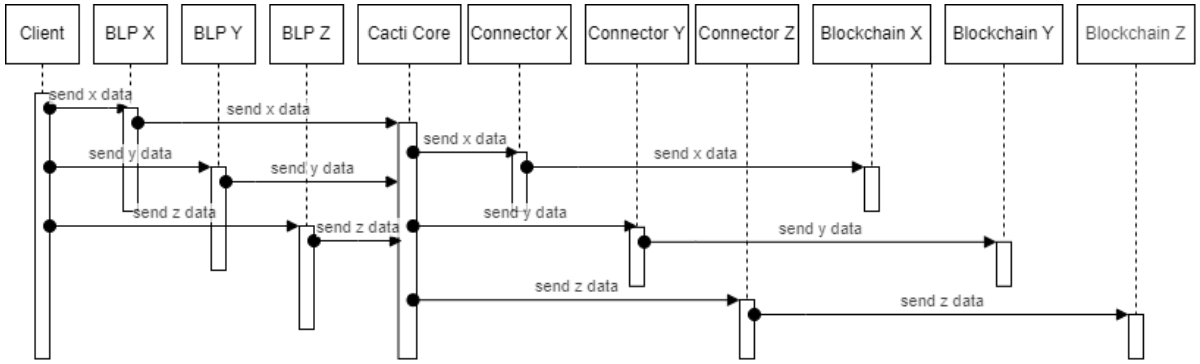

Figure 3.1: UML sequence diagram of an architecture supported by Hyperledger Cacti

3.2 Supply Chain Use Case

The use case chosen for the POC focuses on the supply chain domain. This choice comes from the fact that this business sector inherently utilizes ambiguous terms to describe processes, making

it adequate for demonstrating the value of the proposed solution.

The use case consists of three sequential data transfers among three entities, where each entity leverages a different type of blockchain for business-specific reasons irrelevant to the study. A data transfer is the process of copying information from one DLT to another, and it involves specific data (e.g., manufacturer information) in most cases. This operation can involve multiple steps, including an optional intermediate processing step that may manipulate or analyze the data before sending it to the target chain. Data transfers are vital for ensuring the interoperability and exchange of valuable information across blockchains [BRH+23].

The data model comprises a business role, function, item, and process. That is, a *Supplier* may *Supply* a *Bamboo Harvest* to a *Manufacturer* in the context of a *Product Sale* and so on. There is an obvious concern for preserving privacy through the entire interoperation process among the three entities, as in most cases, the transferred data includes sensitive business information.

Figure 3.2 depicts a sample flow of transactions taking place among the three entities of the use case as a Business Process Model and Notation (**BPMN**) diagram [Whi04]. To achieve interoperability between the three entities, Cacti is used. When an event occurs in one system, a smart contract is invoked. The Cacti Core then interacts with the appropriate Connector (Fabric, Besu, or Quorum) to update the respective blockchain.

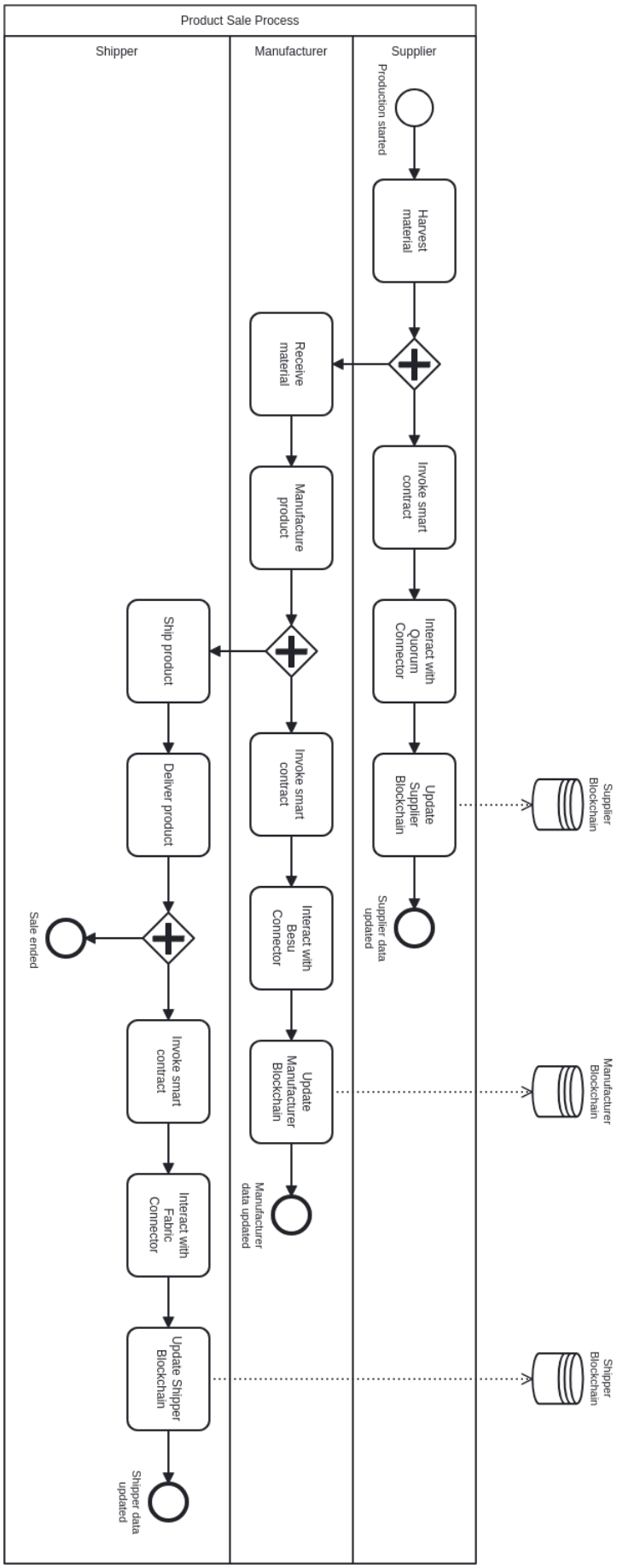

Figure 3.2: Supply chain use case BPMN diagram

Chapter 4

Ontologically-Guided Blockchain Interoperability

This chapter presents a solution for achieving interoperability between two or more blockchain systems by leveraging a common ontology that derives from an EA model. The approach is divided into design-time and runtime activities of the interoperation process to ensure standardization. The former are the activities that take place before the interoperation process has begun and are described in Section 4.1. The latter are the activities that take place during the interoperation process and are presented in Section 4.2. In order to aid the reproduction of this approach, the complete source code of its implementation is published in a public repository.

4.1 Design-Time Activities

At design time, a model is created using Archimate, a modelling language for EA [JLB+16]. This model is the foundation for producing a shared vocabulary as an RDF/XML ontology which represents the structural and conceptual aspects of the Archimate model in a machine-readable format [DMVH+00]. Lastly, the HermiT reasoner is used to verify the extracted ontology [BC14].

4.1.1 Enterprise Architecture Model Design

Ideally, this process is conducted by consortium stakeholders to ensure alignment among participating entities. As they possess rich insights about the meaning of terms and the specific format of messages of each business area, close inter-collaboration determines the scope and contents of the EA model. Therefore, this should be an iterative process, and the model may change significantly as the knowledge grows. Furthermore, the European Interoperability Reference Architecture (**EIRA**) [Com15] should guide the design to ensure regulatory compliance.

Figure 4.1 depicts the three layers of the supply chain-specific EA that was defined in Subsection 3.2. The upper layer portrays the Business layer with the relations between the key business entities. Those are the business roles, functions, processes, and items that will serve as the main study of this thesis. The middle layer represents the Application layer with the software applications that support the business layer. It includes components such as application services and blockchain interfaces. Finally, the bottom layer illustrates the Technology layer with the technological infrastructure that supports the application layer. Its elements are the infrastructure and networks that enable blockchain transactions.

4.1.2 Ontology Extraction and Assessment

After designing the Archimate model, the next step is to extract its model exchange file in XML and convert it into an RDF ontology. A third-party tool called archimate2rdf designed specifically for converting Archimate models into RDF format handles this.

An RDF ontology is favored for managing data from multiple sources because unlike alternative ontology technologies, it accommodates various data types, ensuring seamless integration among diverse systems. Moreover, its semantic richness enables it to capture nuanced meanings and contextual relationships within the information.

Once the RDF ontology is ready, the assessment process using HermiT can proceed. HermiT is a reasoner that can analyze ontologies and infer additional knowledge based on the logical rules defined in the ontology. This process helps identify redundancies and missing information within the ontology.

HermiT can also check for the satisfiability and consistency of the ontology. Satisfiability determines if all the defined classes of the ontology have at least one instance, while consistency checks if there are no logical contradictions within the ontology. These checks are essential to ensure the ontology accurately represents the intended knowledge.

By using HermiT to assess the RDF ontology derived from the Archimate model, one gains valuable insights into the quality of the ontology. This analysis can help uncover potential issues or improvements in the design of the initial model, ensuring that the resulting ontology aligns with the intended architectural knowledge.

The following Listings 4.1, 4.2 and 4.3, display the format of sample business entities in the ontology.

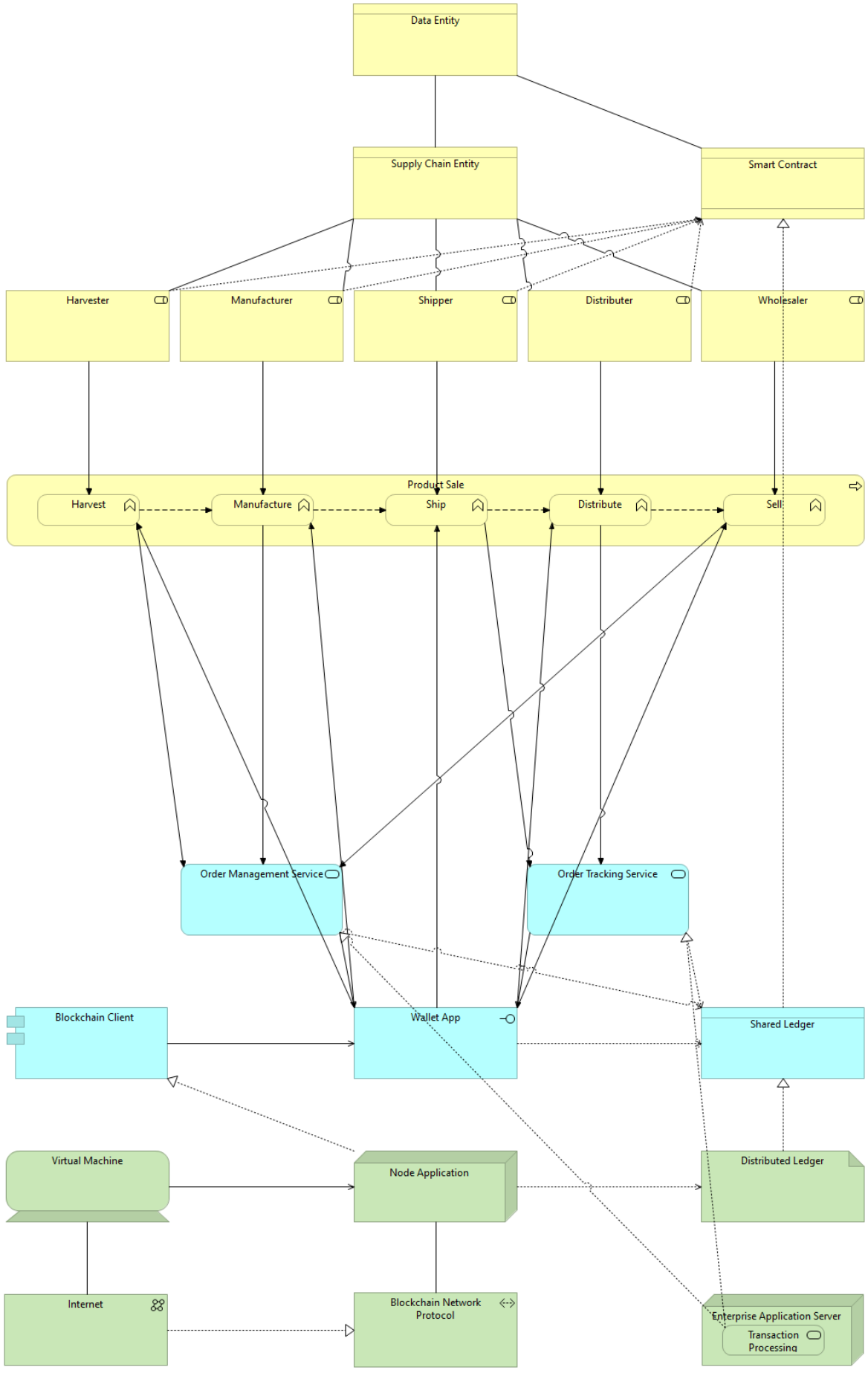

Figure 4.1: Supply chain blockchain interoperability EA model in Archimate

```
<NamedIndividual rdf:about="urn:uuid:id-6d7fc1370fcb48959daae518ea90b265">
    <rdf:type rdf:resource="http://bp4mc2.org/def/archimate#BusinessRole"/>
    <archimate:association
        rdf:resource="urn:uuid:id-67ccb819d7284462a8b9045f23e437d0"/>
    <archimate:triggering
        rdf:resource="urn:uuid:id-00f2f859356543cb8c0b8212e2731e1c"/>
    <archimate:writeAccess
        rdf:resource="urn:uuid:id-a5e7d160476c449ca342489c9b3f3d9f"/>
    <rdfs:label xml:lang="en">Manufacturer</rdfs:label>
</NamedIndividual>
```

Listing 4.1: Sample *Manufacturer* business role entity

```
<NamedIndividual rdf:about="urn:uuid:id-00f2f859356543cb8c0b8212e2731e1c">
    <rdf:type rdf:resource="http://bp4mc2.org/def/archimate#BusinessFunction"/>
    <archimate:flow rdf:resource="urn:uuid:id-94c418f274254d8a920235a645303a6e"/>
    <archimate:triggering
        rdf:resource="urn:uuid:id-01a27c33b42045f58a72188c63a03964"/>
    <archimate:writeAccess
        rdf:resource="urn:uuid:id-98c01165200a4f37872c7b85865a05d4"/>
    <rdfs:label xml:lang="en">Manufacture</rdfs:label>
</NamedIndividual>
```

Listing 4.2: Sample *Manufacture* business function entity

```
<NamedIndividual rdf:about="urn:uuid:id-eda65277dfba4fd987007920fc06210c">
    <rdf:type rdf:resource="http://bp4mc2.org/def/archimate#BusinessProcess"/>
    <archimate:composition
        rdf:resource="urn:uuid:id-00f2f859356543cb8c0b8212e2731e1c"/>
    <archimate:composition
        rdf:resource="urn:uuid:id-561dbe4948034f29b950c9ee130847ff"/>
    <archimate:composition
        rdf:resource="urn:uuid:id-94c418f274254d8a920235a645303a6e"/>
    <archimate:composition
        rdf:resource="urn:uuid:id-e939a980803d4261825d201a2d7d4d83"/>
    <archimate:composition
        rdf:resource="urn:uuid:id-f32e4603611141d899779f82e71be7ba"/>
    <rdfs:label xml:lang="en">Product Sale</rdfs:label>
</NamedIndividual>
```

Listing 4.3: Sample *Product Sale* business process entity

4.2 Runtime Activities

At runtime, the integration process unfolds through a series of steps. First, the transaction context is defined, which consists of stating the target business role and the business process in which the transaction occurs. Afterward, the source system sends the data to the shared Cacti BLP middleware. There, the data is mapped and converted to conform to the ontology. Finally, the BLP validates the data against the predefined constraints and rules. If the data is deemed valid, it is delivered to the target system and subsequently updated in the target blockchain.

4.2.1 Entity Mapping and Conversion

Mapping refers to retrieving the name of a relevant entity from the ontology. It can be either the same entity or a semantically similar one. Conversion refers to building the corresponding ontology role object based on the source role object by extracting all the available data. Both mapping and conversion processes involve querying the ontology in SPARQL by utilizing an open-source library called rdflib.js [TVHVSV18].

The first step is to create an rdflib.js store object that holds the ontology data. This store acts as a container for RDF triples, where each triple consists of a subject, predicate, and object. The store then reads the ontology file and populates itself with the existing triples in the ontology.

SPARQL, a standard query language for RDF data, is used to query the store. rdflib.js provides a convenient API to execute queries on the store. A SPARQL query can specify conditions to filter entities based on specific criteria, such as their properties or relationships with other entities. For example, one can query for all entities of a specific type or entities that possess a particular property value.

4.2.2 Transaction Validation

The transaction validation process consists of two steps. The first is data validation, and the second is state validation. Successful completion of both steps ensures a valid transaction. If either step is unsuccessful, the transaction cannot occur, and the remaining process should abort.

Data validation refers to checking whether the properties of the object are valid. Specifically, it ensures that the values are within the expected boundaries stipulated in the EA model. This step is depicted in Algorithm 1. This algorithm validates data in the ontology role input object by checking specific fields containing the word *Format*. For each of these fields, it ensures a corresponding non-format field exists. If not, it moves to the next field. The code validates the

format field as a regular expression; if invalid, it returns false. It then checks if the non-format field matches the specified pattern. If not, it also returns false. If all checks pass, it returns true.

Algorithm 1: Data Validation Algorithm

```
Input: ontologyRoleObj: Object
Output: isValid: Boolean

Function validateData(ontologyRoleObj)
    fieldsToCompare ← getFieldsContainingWord(ontologyRoleObj, "Format");

    for i ← 0 to length of fieldsToCompare do
        formatFieldName ← fieldsToCompare[i];
        fieldName ← remove "Format" from formatFieldName;

        if fieldName not in ontologyRoleObj then
            continue;
        end

        fieldValue ← ontologyRoleObj[fieldName];
        formatValue ← ontologyRoleObj[formatFieldName];

        if formatValue is not a regular expression then
            return false;
        end

        if fieldValue does not match the format specified in formatFieldName then
            return false;
        end
    end

    return true;
end
```

State validation refers to assuring that the transaction is legal. In other words, a transition from or to specific business functions can occur in the current business process according to the EA model. This step is depicted in Algorithm 2. This algorithm performs a validation check on the input parameters *sourceRole*, *targetRole*, and *process*. It first ensures that *sourceRole* is not falsy; if it is, the method returns false. Then, it attempts to fetch a list of functions related to the specified *process* using an RDF query and returns false if no functions are found. The method then iterates through the list of functions, checking if any of them are triggered by the given *sourceRole* or *targetRole*. The method returns true if *sourceFunction* is not null and either *targetFunction* is not null and the *areAdjacent* method confirms the adjacency of *sourceFunction* and *targetFunction*, or if *targetRole* is falsy.

Algorithm 2: State Validation Algorithm

Input: sourceRole: String, targetRole: String, process: String
Output: isValid: Boolean

Function `validateState(`*sourceRole, targetRole, process*`):`

```
if sourceRole is empty then
    return false;
end
processObj ← RDFQuerier.fetchDetailsByLabel(process);
if processObj is null then
    return false;
end
foreach funcUri in processObj.get('archimate#composition') do
    funcName ← RDFQuerier.fetchLabelByUri(funcUri);
    func ← find function in this.businessFunctions where func.entity equals
     funcName;
    if func.triggeredBy equals sourceRole then
        sourceFunction ← func;
    end
    if func.triggeredBy equals targetRole then
        targetFunction ← func;
    end
    add func to funcs;
end
return (sourceFunction ≠ null) ∧ ((targetFunction ≠ null ∧
 areAdjacent(sourceFunction, targetFunction, funcs)) ∨ ¬targetRole);
```

end

Chapter 5

Evaluation

It is essential to obtain a comprehensive understanding of the advantages and limitations of a novel approach. Conducting both accuracy and performance analyses allows for a rigorous assessment of its capabilities across various dimensions.

Accuracy analysis scrutinizes the reliability of the system irrespective of its operational load. This examination serves the purpose of discerning the inherent capabilities of the system under controlled conditions. It is also pivotal in ensuring that the system effectively fulfills its intended objectives and operates correctly, laying the foundation for a thorough evaluation.

Conversely, performance analysis explores the system's behavior under diverse load conditions, simulating real-world scenarios of varying demand and stress levels. This data enables the assessment of scalability, resilience, and the system's ability to perform consistently in a production environment.

Integrating accuracy and performance analyses is crucial for a holistic evaluation. This method minimizes the risk of oversight and mitigates potential biases. Furthermore, it supports the principle of continuous improvement. As the system evolves, periodic reevaluation through these lenses can assist in addressing emerging challenges.

This chapter begins by describing the utilized environment to evaluate the system in Section 5.1. Sections 5.2 and 5.3 present the accuracy and performance evaluations respectively. Furthermore, this chapter discusses the results and covers the considerations that establish the boundaries within which the solution operates in Sections 5.4 and 5.5.

5.1 Experimental Setup

The experimental infrastructure consisted of three test ledgers (one for each supply chain entity) interoperating with Cacti. The three test ledgers were Besu [FLKM22], Quorum [Hou18],

and Fabric [ABB+18]. These ledgers simulated different blockchain networks and enabled the deployment and interaction with smart contracts. Each component ran on a separate Docker [Doc20] container on a single machine equipped with an AMD Ryzen 7 3750H processor and 16GB of RAM.

The deployment of smart contracts was a critical aspect of the experimental setup, as each of them handled different supply chain aspects. The *BambooHarvestRepository.sol* and *BookshelfRepository.sol* contracts were written in Solidity and the former managed bamboo harvest records, while the latter handled bookshelf records. The *Shipment.ts* script written in Go represented a smart contract for managing shipments. The bamboo harvest and bookshelf smart contracts were deployed in the Besu and Quorum blockchains respectively while the shipment smart contract was deployed on the Fabric ledger.

The deployment process involved creating test accounts, generating contract addresses, and storing contract artifacts in the keychain plugin to manage cryptographic keys securely [HXHB18]. It also included specifying the target organizations, channel, and policy that determines which members of each organization can approve a transaction involving a specific smart contract [ZYH+19].

The purpose of the above setup was to provide a self-contained and local environment that allowed for assessing the effectiveness of the proposed solution by evaluating the POC that puts it into practice. Thus, it provided the necessary functionality for the experiments without requiring a complex and time-consuming process.

5.2 Accuracy Analysis

The chosen metric for the accuracy analysis was the error rate of the POC when validating transactions (Tx) among blockchain systems. Therefore, the approach followed was to conduct experiments with valid and invalid transactions to ascertain if the solution correctly identified each case.

Each validation process scrutinizes two main elements. Those are the transaction context and the object that initiates the transaction. If either of them is invalid, the transaction should not complete.

Specifically, a context is valid when its target role and business process entities are present in the ontology. Similarly, an object is in a legal state when its source role entity exists in the ontology and its fields conform to the specified format.

In the first experiment, both the context and the object were valid. Here the transaction was deemed successful as expected because all steps passed.

The second experiment had an invalid context and valid object. This time, despite the mapping step executing successfully, the validation step failed because of its invalid context.

The third experiment involved a context that was valid and an invalid object. Here, the mapping step was unsuccessful, so the validation failed as it did not even get a chance to execute.

In the fourth experiment, both the context and the object were invalid. Again, the mapping failed, so the transaction validation was deemed unsuccessful.

The experiment results are summarized in Table 5.1.

Experiment#	**Context**	**Object**	**Valid Tx**	**Expected**
1	Valid	Valid	✓	Yes
2	Invalid	Valid	×	Yes
3	Valid	Invalid	×	Yes
4	Invalid	Invalid	×	Yes

Table 5.1: Experiment results

5.3 Performance Analysis

The metric used for the performance analysis was the processing time to complete a transaction among two separate blockchain systems in the POC.

Given that the proposed solution is supported by the infrastructure of Hyperledger Cacti, it is worth noting that the evaluation of the performance of the former is framed to that of the latter.

Experiments were conducted for three distinct groups of "low," "normal," and "high" load profiles corresponding to a rate of 10, 50, and 100 simultaneous transactions. These three load profiles are the most common in an enterprise ecosystem. Furthermore, worker threads permitted the parallel execution of these transaction load groups.

To extract objective and unbiased results, each transaction load group execution was repeated 100 times and their mean time was calculated. While performing the first transactions, the systems may not have reached their processing potential, so the first iterations can be considered a warm-up phase.

Figure 5.1 depicts the impact on processing time as the rate of simultaneous transactions increases before and after integrating the proposed solution. One may interpret the graph as the processing overhead imposed on the participating systems when leveraging the approach presented in this work.

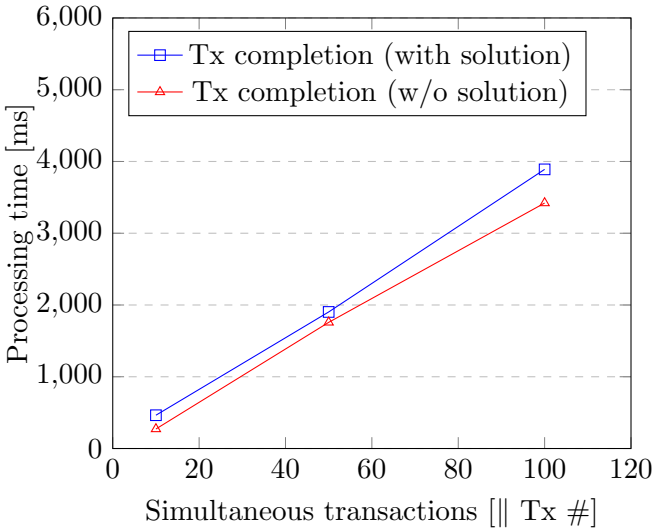

Figure 5.1: Overhead of proposed solution

5.4 Result Discussion

The experiments show that the POC leveraging the proposed solution identifies valid and invalid transactions among systems with 100% accuracy. The benefits of this outcome are immense, as the interoperation process becomes transparent and efficient.

The increase in the processing overhead while utilizing the proposed solution indicates possible scalability issues. Nevertheless, the overhead is considered negligible for enterprise standards until a certain point. Thus, it is suitable for private ecosystem usage where simultaneous transactions typically do not exceed this threshold, and data quality is of foremost concern.

In other words, it is acceptable to sacrifice system performance for reliability. As a result, the capabilities of the individual blockchain systems get enhanced by reducing duplication of effort, improving data integrity and consistency, and enabling new decentralized applications and services to be built on top of them securely.

Another observation is the low cost and ease of extensibility of the solution. While the design and consequent change management of modifying the EA model involve resources, the possibility of automation of the remaining processes implies that spending does not increase for those.

The applicability of the proposed solution is ample. Every network that involves complex transactions among participating systems would gain from this solution. Thus, this approach could potentially impact many industries that actively leverage blockchains and, perhaps, encourage others to migrate their current infrastructure to DLT.

5.5 Considerations

This section discusses the constraints and assumptions that shaped the implementation, providing insights into the challenges involved in achieving standardized data exchange and collaboration across blockchain systems through a shared ontology.

A fundamental constraint of the solution lies in the need to establish a common ontology. This requires consensus and collaboration between the blockchain systems involved. The ontology must define the concepts, relationships, and properties relevant to the specific domain in which the systems operate. However, reaching an agreement on this common ontology may present challenges, as different systems may have varying perspectives and terminology.

Another significant constraint is the availability of transaction context. The solution assumes that information regarding the target entity and the process in which the transaction takes place is accessible. This contextual knowledge is crucial for accurately validating and ensuring the integrity of transactions. However, obtaining and maintaining this contextual information may introduce complexity and overhead, particularly in decentralized and dynamic blockchain ecosystems.

Data conversion and mapping are essential steps for achieving interoperability. The solution assumes that a clear mapping can be established between the concepts and properties of the source blockchain system and the common ontology. This mapping process is critical for seamless data exchange between the systems. However, mapping complex and heterogeneous data structures can be non-trivial, requiring careful analysis and consideration of data semantics and structural differences.

Successful data conversion leads to the subsequent step of data validation. The solution assumes that the converted data can be validated against the constraints and rules defined within the common ontology. This validation ensures that the data adheres to the ontology's specifications and maintains its integrity throughout the interoperability process. Nonetheless, validating data against complex ontologies can be computationally intensive, particularly when dealing with large-scale and dynamic systems.

The POC introduces additional assumptions to simplify the implementation. It assumes that all participating entities utilize the same backend and frontend instances, reducing complexity for demonstration purposes. However, in real-world scenarios, entities would typically have their own independently developed backend and frontend instances, necessitating additional considerations for achieving interoperability.

Furthermore, the POC assumes trivial mapping and conversion steps due to the common backend model shared by all entities involved. This assumption alleviates the ambiguity that

may arise when models are defined separately. In real-world scenarios, where different entities may have distinct models, the mapping and conversion processes would require careful handling of potential conflicts and inconsistencies.

The solution assumes validation of the business process defined in the EA. It expects a sequential execution of functions triggered by the source and target entities, with the latter function succeeding the former. However, ensuring adherence to the defined business process introduces challenges, as it requires coordination and synchronization between the participating systems.

Additionally, the solution assumes fixed transactions among three clearly-defined business entities for demonstration purposes. Therefore, it is limited in scope and does not apply directly to other use cases. To extend the solution's applicability, code generation techniques would be necessary to adapt it to different scenarios and enable broader interoperability [SLS18].

Understanding that some of these aspects are inherent to the approach followed in this solution is crucial for realizing its potential benefits and drawbacks. However, others are imposed by the limited time to develop the POC, and additional resources can trivially overcome them.

Chapter 6

Conclusion

There are a plethora of solutions related to achieving interoperability in blockchain systems. Nevertheless, most of them do not focus on the semantic layer of interoperability. This thesis explored this gap by implementing a solution that enhances interoperation among systems owned by members of a supply chain consortium, as those usually entail operations among different business domains, demonstrating the potential of this mechanism. The novel approach leverages EA modeling to provide a common ground for systems to achieve interoperability in the semantic layer. It consists of an ontologically guided business logic plugin that validates cross-chain transactions. The outcome of the evaluation is satisfactory, proving the added value of this approach. Although the POC is limited to a network of permissioned ledgers owned by a business consortium, the paradigm can be extended for a set of blockchain systems of indefinite size. Thus, this thesis supports that EA techniques can aid in solving open issues in BI and, in this way, encourage further research in this area. As mentioned before, this work needs to be generalized for ease of extensibility to other use cases. For instance, applying this mechanism to enhance interoperability amongst public chains in fields other than supply chain management would be noteworthy. Another interesting future research path is to extend the Hermes fault-tolerant middleware and Hephaestus, solutions described in the preliminary section of this document. The goal would be to increase their robustness by leveraging the transaction validation capabilities offered by the proposed solution.

Bibliography

[ABB+18] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.

[ACB+13] Goncalo Antunes, Artur Caetano, Marzieh Bakhshandeh, Rudolf Mayer, and José Borbinha. Using ontologies for enterprise architecture model alignment. In *Proceedings of the 4th Workshop on Business and IT Alignment (BITA 2013). Poznan, Poland*, 2013.

[AHI+21] Ermyas Abebe, Yining Hu, Allison Irvin, Dileban Karunamoorthy, Vinayaka Pandit, Venkatraman Ramakrishna, and Jiangshan Yu. Verifiable observation of permissioned ledgers. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2021.

[ARK19] ARK. Ark whitepaper. 2019.

[B+14] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[BC14] Ritika Bansal and Sonal Chawla. An approach for semantic information retrieval from ontology in computer science domain. *International Journal of Engineering and Advanced Technology (IJEAT)*, 4(2):58–65, 2014.

[BRH+23] Rafael Belchior, Luke Riley, Thomas Hardjono, André Vasconcelos, and Miguel Correia. Do you need a distributed ledger technology interoperability solution? *Distributed Ledger Technologies: Research and Practice*, 2(1):1–37, 2023.

[Bro18] Richard Gendal Brown. The corda platform: An introduction. *Retrieved*, 27:2018, 2018.

[BSF+19] Michael Borkowski, Marten Sigwart, Philipp Frauenthaler, Taneli Hukkinen, and Stefan Schulte. Dextt: Deterministic cross-blockchain token transfers. *IEEE access*, 7:111030–111042, 2019.

[BSP+22] Rafael Belchior, Peter Somogyvari, Jonas Pfannschmid, André Vasconcelos, and Miguel Correia. Hephaestus: Modelling, analysis, and performance evaluation of cross-chain transactions. 2022.

[BTP+22] R Belchior, L Torres, J Pfannschmid, A Vasconcelos, and M Correia. Can we share the same perspective? blockchain interoperability with views, 2022.

[But16] Vitalik Buterin. Chain interoperability. *R3 Research Paper*, 9, 2016.

[BVCH22] Rafael Belchior, André Vasconcelos, Miguel Correia, and Thomas Hardjono. Hermes: Fault-tolerant middleware for blockchain interoperability. *Future Generation Computer Systems*, 129:236–251, 2022.

[BVGC21] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)*, 54(8):1–41, 2021.

[CCA+21] Kevin Coutinho, Ponnie Clark, Ferdinand Azis, Norman Lip, and Josh Hunt. Enabling blockchain scalability and interoperability with mobile computing through layerone. x. *arXiv preprint arXiv:2110.01398*, 2021.

[Cha] Chatterjee. Rosetta build once. integrate your blockchain everywhere.

[CM18] Arlyn Culwick and Dan Metcalf. Building super financial markets for the new digital economy. 2018.

[Com15] European Commission. Eira support series - how eira supports interoperability v1.0.0, 2015.

[CZDK17] ZD Chen, Y Zhuo, Zhang-Bo Duan, and H Kai. Inter-blockchain communication. *DEStech Transactions on Computer Science and Engineering*, 2017.

[Dan17] Chris Dannen. *Introducing Ethereum and solidity*, volume 1. Springer, 2017.

[DKKZ20] Stelios Daveas, Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros. A gas-efficient superlight bitcoin client in solidity. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 132–144, 2020.

[DMVH+00] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of xml and rdf. *IEEE Internet computing*, 4(5):63–73, 2000.

[Doc20] Inc Docker. Docker. *lnea].[Junio de 2017]. Disponible en: https://www. docker. com/what-docker*, 2020.

[ERH19] Tumennast Erdenebold, Jae Jeung Rho, and Yoon Min Hwang. Blockchain reference model and use case for supply chains within enterprise architecture. *Journal of Information Technology and Architecture*, 16(1):1–10, 2019.

[FBP20] Enrique Fynn, Alysson Bessani, and Fernando Pedone. Smart contracts on the move. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 233–244. IEEE, 2020.

[FLKM22] Caixiang Fan, Changyuan Lin, Hamzeh Khazaei, and Petr Musilek. Performance analysis of hyperledger besu in private blockchain. In *2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 64–73. IEEE, 2022.

[Fus17] Fusion. An inclusive and cryptofinance platform and based on blockchain. 2017.

[G+15] Gideon Greenspan et al. Multichain private blockchain-white paper. 2015.

[GKO20] Alberto Garoffolo, Dmytro Kaidalov, and Roman Oliynykov. Zendoo: A zk-snark verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1257–1262. IEEE, 2020.

[GRG+21] Bishakh Chandra Ghosh, Venkatraman Ramakrishna, Chander Govindarajan, Dushyant Behl, Dileban Karunamoorthy, Ermyas Abebe, and Sandip Chakraborty. Decentralized cross-network identity management for blockchain interoperation. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2021.

[Her18] Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM symposium on principles of distributed computing*, pages 245–254, 2018.

[HLP19] Thomas Hardjono, Alexander Lipton, and Alex Pentland. Toward an interoperability architecture for blockchain autonomous systems. *IEEE Transactions on Engineering Management*, 67(4):1298–1309, 2019.

[HLS22] Maurice Herlihy, Barbara Liskov, and Liuba Shrira. Cross-chain deals and adversarial commerce. *The VLDB journal*, 31(6):1291–1309, 2022.

[Hou18] Christopher Hounsom. Quorum whitepaper v0.2. 2018.

[HXHB18] Yifei Hu, Yan Xiong, Wenchao Huang, and Xianglin Bao. Keychain: blockchain-based key distribution. In *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*, pages 126–131. IEEE, 2018.

[JLB+16] Andrew Josey, Marc Lankhorst, Iver Band, Henk Jonkers, and Dick Quartel. An introduction to the archimate® 3.0 specification. *White Paper from The Open Group*, 2016.

[KB19] Jae Kwon and Ethan Buchman. Cosmos whitepaper. *A Netw. Distrib. Ledgers*, page 27, 2019.

[KC19] Victoria Kalogirou and Yannis Charalabidis. The european union landscape on interoperability standardisation: status of european and national interoperability frameworks. In *Enterprise Interoperability VIII: Smart Services and Business Impact of Enterprise Interoperability*, pages 359–368. Springer, 2019.

[KK19] Angelina Kouroubali and Dimitrios G Katehakis. The new european interoperability framework as a facilitator of digital transformation for citizen empowerment. *Journal of biomedical informatics*, 94:103166, 2019.

[Lee18] James Lee. Komodo: An advanced blockchain technology, focused on freedom. *Komodo Platform, Komodo*, 12, 2018.

[LXS+19] Zhuotao Liu, Yangxi Xiang, Jian Shi, Peng Gao, Haoyu Wang, Xusheng Xiao, Bihan Wen, and Yih-Chun Hu. Hyperservice: Interoperability and programmability across heterogeneous blockchains. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 549–566, 2019.

[MBPH+20] Hart Montgomery, Hugo Borne-Pons, Jonathan Hamilton, Mic Bowman, Peter Somogyvari, Shingo Fujimoto, Takuma Takeuchi, Tracy Kuhrt, and Rafael Belchior. Hyperledger cactus whitepaper. 2020.

[MMSS+18] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. Anonymous multi-hop locks for blockchain scalability and interoperability. *Cryptology ePrint Archive*, 2018.

[Nak08] Satoshi Nakamoto. Bitcoin whitepaper. 2008.

[NSER20] Sergey Nazarov, Punit Shukla, A Erwin, and A Rajput. Bridging the governance gap: Interoperability for blockchain and legacy systems. In *World Economic Forum whitepaper*, 2020.

[Pan21] Ramakrishna Pandit. Blockchain interoperability: Challenges, ongoing efforts, and potential solutions, 2021.

[Pec17] Morgen E Peck. Blockchains: How they work and why they'll change the world. *IEEE Spectrum*, 54(10):26–35, 2017.

[Pet20] Niklas Petersen. *Towards Semantic Integration of Supply Chain and Production Data*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2020.

[SLBS20] Kevin Sekniqi, Daniel Laine, Stephen Buttolph, and Emin Sirer. Avalanche platform, 2020.

[SLS18] Eugene Syriani, Lechanceux Luhunu, and Houari Sahraoui. Systematic mapping study of template-based code generation. *Computer Languages, Systems & Structures*, 52:43–62, 2018.

[ST+17] Matthew Spoke, NE Team, et al. Aion: Enabling the decentralized internet. *AION, White Paper*, 2017.

[TVHVSV18] Ruben Taelman, Joachim Van Herwegen, Miel Vander Sande, and Ruben Verborgh. Comunica: a modular sparql query engine for the web. In *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II 17*, pages 239–255. Springer, 2018.

[Und16] Sarah Underwood. Blockchain beyond Bitcoin. *Communications of the ACM*, 59(11):15–17, 2016.

[VGJ+16] David Voell, Frank Lu-Nick Gaski, Ram Jagadeesan, Renat Khasanshyn, Hart Montgomery, Stefan Teis, Tamas Blummer, Murali Krishna Katipalli, and Mic Bowman. Hyperledger whitepaper. 2016.

[VTPM18] Gilbert Verdian, Paolo Tasca, Colin Paterson, and Gaetano Mondelli. Quant overledger whitepaper. *Release V0. 1 (alpha)*, 2018.

[Wan22] Wanchain. Wanchain whitepaper. 2022.

[Whi04] Stephen A White. Introduction to bpmn. *Ibm Cooperation*, 2(0):0, 2004.

[Woo16] Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework. *White paper*, 21(2327):4662, 2016.

[ZHL+19] Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, and William Knottenbelt. Xclaim: Trustless, interoperable, cryptocurrency-backed assets. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 193–210. IEEE, 2019.

[ZYH+19] Wenbin Zhang, Yuan Yuan, Yanyan Hu, Karthik Nandakumar, Anuj Chopra, Sam Sim, and Angelo De Caro. Blockchain-based distributed compliance in multinational corporations' cross-border intercompany transactions: A new model for distributed compliance across subsidiaries in different jurisdictions. In *Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication Conference (FICC), Vol. 2*, pages 304–320. Springer, 2019.