INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Blockchain and Property Technology

DANIEL FILIPE MERÊNCIO DIAS

Outubro de 2023

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Blockchain and Property Technology

Daniel Dias

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Graphic Systems and Multimedia

Supervisor: David Mota
Advisor: Paulo Gandra de Sousa

Evaluation Committee:
President:

Members:

Porto, October 14, 2023

Abstract

Blockchain technology has emerged as a revolutionary force in the real estate sector, promising transformative changes in how properties are bought, sold, and managed. This study delves into the integration of blockchain in real estate, emphasizing the development and testing of a practical proof-of-concept application. The research, undertaken during an internship at DevScope, critically examines the potential of blockchain to enhance transparency, security, and efficiency in managing exclusivity contracts.

The research starts with a comprehensive review of blockchain literature, exploring its diverse applications within the real estate industry. It delves into the intricacies of challenges faced and opportunities presented during the implementation of blockchain technology in this context. Through rigorous analysis, the study assesses the profound impact of blockchain on the real estate landscape, elucidating the multifaceted benefits and challenges inherent in deploying blockchain-based solutions.

Furthermore, this research not only contributes valuable insights to the ongoing discourse surrounding blockchain in real estate but also presents practical implications. The findings are poised to inform strategic decisions, providing DevScope with a nuanced perspective on integrating blockchain technology into their products. Specifically, this study evaluates the feasibility of incorporating a blockchain-based solution into MaxWork, one of DevScope's products, thereby paving the way for innovative advancements within the realm of Multiple Listing Service solutions.

Keywords: Blockchain, Smart Contracts, Real Estate, Multiple Listing Service, Distributed Ledger, Immutability

Resumo

A tecnologia Blockchain emergiu como uma força revolucionária no setor imobiliário, prometendo mudanças na forma como as propriedades são transacionadas e geridas. Este estudo explora a integração da blockchain no setor imobiliário, a partir do desenvolvimento de uma aplicação de prova de conceito. A pesquisa, realizada durante um estágio na DevScope, examina o potencial da Blockchain para melhorar a transparência, segurança e eficiência nas transações imobiliárias.

A pesquisa inicia-se com uma revisão da literatura sobre Blockchain e Distributed Ledger Technology (DLT), explorando as diversas aplicações dentro da indústria imobiliária. O estudo avalia o impacto da blockchain no cenário imobiliário, elucidando os benefícios e desafios inerentes à implementação de soluções baseadas em Blockchain.

Além disso, esta pesquisa não só contribui com insights valiosos para o contínuo debate sobre Blockchain no setor imobiliário, como também apresenta implicações práticas. As descobertas deste estudo proporcionam à DevScope uma perspectiva detalhada sobre a integração da tecnologia Blockchain. Especificamente, este estudo avalia a viabilidade de incorporar uma solução baseada em Blockchain no MaxWork, um dos principais produtos da DevScope, abrindo caminho para avanços inovadores no âmbito dos Multiple Listing Services (MLS).

Contents

List of Figures ix

List of Tables xi

List of Acronyms xiii

List of Source Code xv

1 1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Problem . . . 1
- 1.3 Objectives . . . 2
- 1.4 Research Methodology . . . 3
- 1.5 Document Structure . . . 3

2 Background 5
- 2.1 Multiple Listing Service (MLS) . . . 5
- 2.2 Distributed Ledger Technology (DLT) . . . 5
- 2.3 DLT Technologies . . . 7
  - 2.3.1 Blockchain . . . 7
  - 2.3.2 Tangle . . . 9
  - 2.3.3 Hashgraph . . . 10
  - 2.3.4 Summary . . . 11
- 2.4 Blockchain in-depth analysis . . . 12
  - 2.4.1 Is Blockchain trust-free? . . . 14
  - 2.4.2 Taxonomy of Blockchain Systems . . . 14
  - 2.4.3 Consensus mechanisms . . . 15
    - Proof of Work (PoW) . . . 16
    - Proof of Stake . . . 16
    - Practical Byzantine Fault Tolerance . . . 16
    - Raft . . . 16

3 State of the Art 19
- 3.1 Generations of blockchain technology . . . 19
    - Blockchain 1.0 . . . 19
    - Blockchain 2.0 . . . 19
    - Blockchain 3.0 . . . 22
- 3.2 Real-World Applications of Blockchain in Real Estate . . . 23
  - 3.2.1 Enterprise solutions . . . 23
  - 3.2.2 Countries/Cities . . . 23
- 3.3 Blockchain Technology Platforms . . . 25

3.3.1 Hyperledger Fabric . . . 25
3.3.2 R3 Corda . . . 26
3.3.3 Quorum . . . 26
3.3.4 MultiChain . . . 27

4 Value Analysis **29**
4.0.1 Innovation process . . . 29
4.0.2 Opportunity Identification . . . 31
4.0.3 Opportunity Analysis . . . 31
4.0.4 Idea Generation . . . 32
4.0.5 Idea Selection . . . 33
4.0.6 Analytic Hierarchy Process (AHP) . . . 33
4.0.7 Technique of Order Preference by Similarity to Ideal Solution (TOPSIS) . . . 36
4.0.8 Concept and Technology Development . . . 39
4.1 Value . . . 39
4.1.1 Value Proposition . . . 41
4.1.2 Quality Function Deployment . . . 42

5 Solution **45**
5.1 Software Requirements Specification . . . 45
5.1.1 Actors . . . 45
5.1.2 Functional Requirements . . . 45
5.1.3 Non-functional requirements . . . 47
5.2 Domain Modeling . . . 47
5.3 Hyperledger Fabric Network Design . . . 49
5.4 Architectural Design . . . 50
5.4.1 Solution Architecture Overview . . . 50
5.4.2 Deployment Process . . . 51
5.4.3 Implementation . . . 54
5.4.4 Hyperledger Fabric network implementation . . . 55
5.4.5 API implementation . . . 58
5.4.6 Frontend Implementation . . . 59

6 Tests & Solution Evaluation **61**
6.0.1 Testing Methodology for Blockchain Applications . . . 61
6.0.2 Performed Tests . . . 62
6.0.3 Solution Evaluation . . . 64

7 Conclusion **65**
7.1 Project Synopsis . . . 65
7.2 Contributions . . . 66
7.3 Future Work . . . 66

Bibliography **67**

List of Figures

2.1 Blockchain Blocks 7
2.2 Blockchain’s Block Structure (Zheng, S. Xie, H. Dai, et al. 2017) 8
2.3 Blockchain and Tangle structure comparison (Zhang et al. 2022) 9
2.4 Hashgraph illustration example (L. Baird 2016) 10
2.5 Blockchain Overview (Ismail and Materwala 2019) 13
2.6 Types of Blockchain (Ismail and Materwala 2019) 15

4.1 Innovation Process 29
4.2 The New Concept Development Model (NCD) 30
4.3 SWOT Analysis 32
4.4 AHP Hierarchy Tree 34
4.5 Value Proposition Canvas 42
4.6 House of Quality 44

5.1 Use Case Diagram 46
5.2 Domain Model 48
5.3 Hyperledger Fabric high level transaction flow (Androulaki et al. 2018) . . . 50
5.4 PoC Network Architecture 51
5.5 Complete application deployment diagram 52
5.6 Minikube Dashboard 53
5.7 System components architecture 54
5.8 Hyperledger Explorer Dashboard 54
5.9 Agency1 organization definition in configtx.yaml 55
5.10 Profiles section in configtx.yaml 56
5.11 Chaincode class diagram 57
5.12 Register user endpoint 58
5.13 Create asset endpoint 59
5.14 Create Asset chaincode call 59
5.15 Contract list view 59
5.16 Update a contract form view 60
5.17 Update method HTTP call 60

6.1 Create asset test 62
6.2 Exclusivity Contract instance in CouchDB 63
6.3 Hyperledger Explorer transaction details 64

List of Tables

2.1 DLT Comparison 12

4.1 Saaty's Scale 34
4.2 Comparison Matrix 35
4.3 Normalized matrix and relative priority 35
4.4 IR value table 36
4.5 Priority vector matrix 36
4.6 Decision Matrix 37
4.7 Normalized Matrix 37
4.8 Matrix with r_{ij} values 38
4.9 Ideal solution and ideal negative solution 38
4.10 Separation from the ideal and ideal negative solution 39
4.11 Calculation of the proximity to the ideal solution 39
4.12 Benefits and Sacrifices to the customer 41

5.1 Requirement 1 - Submit contract 46
5.2 Requirement 2 - Modify Exclusivity Contract 46
5.3 Requirement 3 - Cancel Exclusivity Contract 47
5.4 Requirement 6 - Verify Ledger 47
5.5 Non-Functional Requirements 47

List of Acronyms

AHP	Analytic Hierarchy Process.
API	Application Programming Interface.
BaaS	Blockchain as a Service.
BFT	Byzantine Fault Tolerance.
CI	Consistency Index.
CR	Consistency Ratio.
CRUD	Create, Read, Update, and Delete.
CTO	Chief Technology Officer.
DAG	Directed Acyclic Graph.
DAO	Decentralized Autonomous Organization.
dApp	Decentralized Application.
DeFi	Decentralized Finance.
DLT	Distributed Ledger Technology.
DPoS	Delegated Proof of Stake.
DSRM	Design Science Research Methodology.
ETC	Ethereum Classic.
ETH	Ethereum.
FSI	Financial Services Industry.
FTP	File Transfer Protocol.
HOQ	House of Quality.
HTTP	Hypertext Transfer Protocol.
IBM	International Business Machines.
IoT	Internet of Things.
IP	Internet Protocol.
IS	Information System.
MCDM	Multi Criteria Decision-Making.
MLS	Multiple Listing Service.
MSP	Membership Service Provider.
MVC	Model-View-Controller.
NAPR	National Agency of the Public Registry.
NFS	Network File System.
NFT	Non-Fungible Token.

NPD	New Product Development.
NPI	Número de Identificação Predial.
pBFT	Pratical Byzantine Fault Tolerance.
PhD	Doctor of Philosophy (*philosophiae doctor*).
PoD	Proof of Deposit.
PoET	Proof of Elapsed Time.
PoS	Proof of Stake.
PoW	Proof of Work.
QFD	Quality Function Deployment.
REIT	Real Estate Funds.
REST	Representational State Transfer.
RI	Random Consistency Index.
RPC	Remote Procedure Calls.
SMTP	Simple Mail Transfer Protocol.
SWOT	(Strengths,Weaknesses,Opportunities,Threats).
TCP	Transmission Control Protocol.
TLS	Transport Layer Security.
TPS	Transactions per second.
UK	United Kingdom.
UML	Unified Modeling Language.
USA	United States of America.
UX	User Experience.

List of Source Code

3.1 Smart contract logical example by Nick Szabo (Szabo 1996) 20

Chapter 1

1 Introduction

The purpose of this document is to assess and evaluate the application of Distributed Ledger Technology (DLT) as a means to establish a more resilient and fortified system for handling contracts specific to the real estate industry.

This chapter provides an overview of the project, beginning with a discussion of the thesis' inherent context, the problem addressed, and the project's objectives. Following that, the research methodology used to carry out this project is described. Finally, the report's structure is defined and chapters are presented succinctly.

1.1 Context

DevScope is responsible for the management of a Multiple Listing Service called MaxWork, which is used by the real estate agency Remax.

A Multiple Listing Service (MLS) is a database created by cooperating real estate brokers to provide information about available properties (Chen 2022). An MLS allows collaboration between brokers to analyze listed properties in order to connect homebuyers and sellers.

This platform also houses transaction records and the exchange of exclusivity contracts, a procedure known in real estate mediation. The real estate mediation process, usually provided by what's called a broker, grants clients a trustworthy pipeline throughout the process of selling or rental of a property. An exclusivity contract consists of an agreement celebrated between the owner of a property and a real estate agency, in order to the latter promote and mediate the sale or the renting process, exclusively, during the time period agreed upon. This means that the property owner can't make a sale on his own without having to compensate the real estate agency according to the terms defined in the contract (Mação 2022).

1.2 Problem

A significant portion of the digitized information is hosted on disparate systems, which results in a lack of transparency, efficiency and a higher incidence of inaccuracies that create a greater potential for fraud. There is still a lot of improvement that can be made in real estate regarding the use of digital technology and the representation of physical assets in digital forms.

The data registry in real estate has always been subject to possible hacker attacks, employee abuse, and all the other negative aspects of centralized databases.

Also, a contract detained by the company that detains the contract and the storing solution is not necessarily viable. The company can manipulate contract data and the software that manages it. DevScope manages Maxwork, an MLS platform that contains real estate data such as properties, agents, contracts, and more. This means that the company is responsible for the storage and authenticity of private and sensitive information. At this time, it's undeniable that DevScope (accountable for the data storage management) cannot be held liable for any alterations to exclusivity contract data.

For an industry like real estate, where various players collaborate in order to complete a sale or reach some type of agreement, the involvement of various agents, agencies, or entities can be beneficial. When this type of collaboration is needed, mechanisms like Distributed Ledger Technologies (DLT) are presented as a possible solution for the creation of more reliable, secure, and trusted systems.

Blockchain is a type of distributed ledger technology that stores all transactions records in a transparent and immutable manner. Once something has been recorded on the blockchain ledger, it can never be changed - the data is secure and unchangeable (Lenz 2019). The implementation of smart contracts was introduced in Ethereum's blockchain. Because smart contracts are strictly algorithmic and cryptographically based, they ensure transaction security and eliminate the risk of ambiguous interpretation of conditions(Fernández-Caramés and Fraga-Lamas 2018). In the case of exclusivity contracts, once an exclusivity contract was signed, the smart contract would trigger the registration of the details of the contract in the ledger. The parties involved in these contracts would benefit from the blockchain ledger's immutability and tamper-evident nature. If someone wanted to change some aspects of the contract, the other parties involved would current on that modification.

.

1.3 Objectives

The main goal of this dissertation is to expose the different distributed ledger technologies that could be used. Then, through the development of a Proof-of-Concept with a focus on contract record-keeping and smart contract implementation, evaluate how it can be a valuable asset to the real estate industry. To achieve the main objective the project will be divided into three stages:

1. The first stage consists of reuniting with the representative of DevScope to gather the requirements information.

2. The second stage entails conducting extensive research on existing platforms based on Distributed Ledger Technologies (Blockchain, Directed Acyclic Graph, Hashgraph) to determine which of these options is more reliable in meeting the protocol's requirements. Furthermore, these two stages are critical for understanding which technology is more appropriate for selecting which platforms best meet the needs of this business context.

3. The third and final stage of this project involves prototyping the application of distributed ledger technology based on the results obtained in the previous steps.

1.4 Research Methodology

Design Science Research Methodology (DSRM) is a well-known and accepted framework to implement design science research in Information Systems (IS) research. This methodology will be used in the development of this dissertation's work and it's composed of six steps (Peffers et al. 2007):

1. Problem identification and motivation: definition of the problem and justification of the value of a solution. Accepting the results after a reasonable understanding of the problem. This dissertation problem is described in section 1.2, and the value analysis is shown in the first chapter and detailed in chapter 4.
2. Definition of the objectives which can be seen in section 1.3. Knowledge gathering of the state of the problem and other solutions can be seen in Chapters 2 & 3.
3. Design and development of the prototype after gathered all requirements. The design and implementation of the solution are described in chapter 5.
4. Demonstration: prove that the artefact can solve the problem or parts of it. It is usually accomplished through experimentation, simulation, case study or similar activities.
5. Efficiency evaluation comparing the objectives to the observed results. The evaluation of this document's solution is presented in chapter 5, section 5.1.1.

Platforms like *ResearchGate* and *Ieeexplore* were used with the identified keywords to find papers about the subject. Papers published within the last three years were prioritized over elder research. The main keywords were:

- blockchain
- distributed ledger
- smart contracts
- decentralization
- real estate

1.5 Document Structure

This document is structured as follows:

- **Introduction**. The initial chapter covers this dissertation's context, along with the description of the problem and its main objectives, followed by the adopted methodology for the research and implementation stages.
- **Background**. This chapter is divided into two parts. The first one provides a theoretical explanation of what distributed ledger technology is along with an overview of its existing types, benefits, and obstacles in order to address why blockchain is the most suitable technology for this context. The second part covers a more in-depth explanation of blockchain.
- **State of the Art**. This chapter will provide a review of the concepts related to this dissertation problem as well as some examples of the use of blockchain technology in real estate.

- **Value Analysis**. This chapter will offer a thorough examination of the value proposition of the proposed solution. By employing diverse methods and models, the chapter will delve into the economic, operational, and strategic benefits of the solution. Moreover, this section will delineate the pivotal moment where the most fitting blockchain platform, in this case, Hyperledger Fabric, is chosen. Through a rigorous evaluation process, the chapter will demonstrate how the chosen platform aligns seamlessly with the project's objectives and significantly enhances the solution's overall value.
- **Solution**. This chapter will offer a comprehensive exploration of the use cases, delve into the architectural intricacies of the proof of concept, and provide a detailed account of the implementation process.
- **Tests and Solution Validation**. This chapter will provide an analysis of the testing methodologies employed to validate the developed solution. Unit tests, integration tests, and end-to-end tests will be conducted to assess the system's functionality, security, and scalability.
- Ultimately, the **Conclusions** chapter encapsulates the essence of the work conducted. It provides a concise summary of the research, outlining its core findings and contributions. Additionally, this section serves as a gateway to the future, delineating the logical progression of the study and highlighting the forthcoming steps and areas of exploration.

Chapter 2

Background

This chapter introduces the concept of MLS and DLT. It also gives a summary of DLT technologies and an in-depth blockchain section.

In this chapter, we delve into the fundamental concepts of Multiple Listing Service (MLS) and Distributed Ledger Technology (DLT). Beginning with an exploration of MLS, and its significance in the real estate industry. Subsequently, the focus shifts to DLT, giving an overview of DLT technologies along with an extensive and detailed analysis of blockchain.

2.1 Multiple Listing Service (MLS)

A MLS is a piece of software used by brokers and real estate agents with the primary objective of disseminating information from listing brokers to other member brokers about houses listed for sale.

The first form of property sharing goes back to the late 1800s when real estate brokers regularly gathered at the offices of their local associations to share information about properties they were trying to sell. If the brokers contributed to the sale of a property they would have the right to compensation and that's how the first MLS was born.Today, brokers share information on properties they have listed and invite other brokers to cooperate in their sale in exchange for compensation if they produce the buyer. Sellers benefit by increased exposure to their properties and buyers benefit because they can obtain information about all MLS-listed properties while working with only one broker.

These systems level the playing field so that the smallest brokerage in town can compete with the biggest multi-state firm. Buyers and sellers can work with the professional of their choice, confident that they have access to the largest pool of properties for sale in the marketplace. The real estate market is competitive, and the business is unique in that competitors must also cooperate with each other to ensure a successful transaction.

One of the fundamental principles of an MLS system is cooperation between agents and due to that need, this type of system would benefit from the integration of a distributed ledger, where the events related to the exclusivity contracts (agent identification, commissions, etc) would be stored in a secure, immutable and distributed way.

2.2 Distributed Ledger Technology (DLT)

A ledger could be defined simply as a database that records transactions in chronological order with the use of a time stamp. (Lenz 2019).

For distributed ledger technology (DLT) there are many different definitions systems in the literature and many publications on the subject set out their own unique definition in their preamble. Some definitions are narrow, while others are very broad; some are contradictory.

For example, Atzori (Atzori 2017) describes it as an "irreversible and tamper-proof public records repository for documents, contracts, properties, and assets [that] can be used to embed information and instructions, with a wide range of applications".

The World Bank (Natarajan, Krause, and Gradstein 2017) describes DLT systems as "a specific implementation of the broader category of "shared ledgers", which are simply defined as a shared record of data across different parties.

In another article, DLT is referred to as "a shared, digital infrastructure for applications on DLT (e.g., in financial transactions) by enabling the operation of a highly available, append-only distributed database (referred to as distributed ledger) in an untrustworthy environment, where separated storage and computing devices maintain a local replication of the ledger."

To Hawlitschek, Notheisen, and Teubner (2018) "DLT enables storing new transactions in a distributed, decentralized network after validation by peers". (Hawlitschek, Notheisen, and Teubner 2018)

As shown by these examples, there is no genuine and universal definition for what is referred to as a DLT system. Unclear terminology and fuzzy boundaries have resulted in 'DLT' evolving into an umbrella term used to designate a variety of loosely related concepts (which include, among others, blockchains). According to Rauchs et al. (Rauchs et al. 2018) a more pragmatic and direct definition would be "A DLT system is a system of electronic records that enables a network of independent participants to establish a consensus around, the authoritative ordering of cryptographically-validated ('signed') transactions. These records are made persistent by replicating the data across multiple nodes and tamper-evident by linking them by cryptographic hashes. The shared result of the reconciliation/consensus process - the 'ledger' - serves as the authoritative version for these records."

However, there's a need to implement certain mechanisms to validate the data in a distributed ledger. Some of them are:

1. Asymmetric private and public digital keys ensure that every new piece of information can be uniquely linked to the sending participant and cannot be changed or manipulated because it is encrypted. (Burkhardt, Werling, and Lasi 2018)
2. An automatically-running software algorithm called *consensus mechanism* guarantees that the same information is only recorded once in the database and the information is not duplicated (for example, the double spending problem) (Burkhardt, Werling, and Lasi 2018)
3. The recorded information is irreversible and immutable recorded within the database by using hash functions and time stamps for new data entries. Any attempt to change the data afterward would destroy the chronological order and logical consistency of the chain of information and would immediately be detected (Chowdhury et al. 2019).
4. The common database has high redundancy because it is kept by multiple network participants (Chowdhury et al. 2019). Therefore, multiple copies exist within the network which is permanently synchronized, so that every network participant has at every time the same information. There is only one single source of truth (or failure) within the network (Chowdhury et al. 2019). The permanent synchronization of data

and the existence of multiple copies makes the database resilient against hacker attacks (Lenz 2019).

This technology is currently being used in many applications such as finance (A. Tapscott and D. Tapscott 2017), smart cities (Shen and Pena-Mora 2018), (J. Xie et al. 2019), supply chain (Benčić, Skočir, and Žarko 2019), (Mondal et al. 2019), public sector (Ølnes, Ubacht, and Janssen 2017), healthcare (Brogan, Baskaran, and Ramachandran 2018), (S. Wang et al. 2018), vehicular network (Shrestha and Nam 2019), and Internet-of-Things (Fernández-Caramés and Fraga-Lamas 2018).

2.3 DLT Technologies

In this section three Distributed ledger technologies will be presented, as well as a summary/comparison and why the chosen technology fits best for this project.

2.3.1 Blockchain

Blockchain technology was developed in the 1990s (Massias, Avila, and QuisquaterUCL 1999) (Haber and W. S. Stornetta 1991) (Bayer, Haber, and W. Stornetta 1999) but only gained importance in 2008 with the invention of the cryptocurrency Bitcoin by Satoshi Nakamoto (Nakamoto 2008).

Blockchain technology was first seen in the cryptocurrency Bitcoin (Nakamoto 2008), where it was used to verify and store transactions of this digital currency. This was nothing short of groundbreaking as it enabled the decentralization of digital value exchange, which was impossible before blockchain technology. Digital transactions before *Bitcoin* and blockchain were always conducted using a trusted third party, or intermediary, such as retail banks. Most of the term blockchain technology refers to the underlying database structure used in these transactions.

Blockchain is a sequence of blocks, which holds a complete list of transaction records like a conventional public ledger (Chuen 2015). Figure 2.1 illustrates an example of a blockchain. With a previous block hash contained in the block header, a block has only one parent block. The first block of a blockchain is called *genesis* block which has no parent block (Chuen 2015).

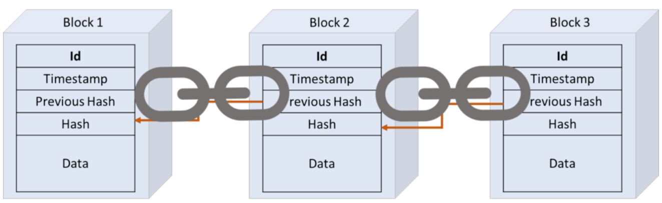

Figure 2.1: Blockchain Blocks
(Paul 2020)

A block consists of the block header and the block body as shown in Figure 2.2. In particular, the block header includes (Zheng, S. Xie, H. N. Dai, et al. 2018):

1. Block version: indicates which set of block validation rules to follow.

2. Merkle tree root hash: the hash value of all the transactions in the block.
3. Timestamp: current time as seconds in the universal time since January 1, 1970 (might vary configuration).
4. nBits: target threshold of a valid block hash.
5. Nonce: a 4-byte field, which usually starts with 0 and increases for every hash calculation.
6. Parent block hash: a 256-bit hash value that points to the previous block.

The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction (Zheng, S. Xie, H. N. Dai, et al. 2018). Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions (NRI 2016).

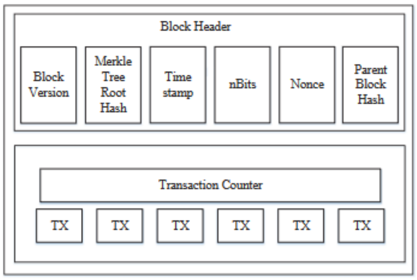

Figure 2.2: Blockchain's Block Structure (Zheng, S. Xie, H. Dai, et al. 2017)

Blockchain possesses the following key characteristics (Zheng, S. Xie, H. Dai, et al. 2017):

- *Decentralization*. Third-party is no longer needed in the blockchain. Consensus algorithms in blockchain are used to maintain data consistency in distributed networks.
- *Persistency*. Transactions can be validated quickly due to having a time-ordered sequence and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or roll back transactions once they are included in the blockchain.
- *Anonymity*. The real identity of blockchain users is not revealed, instead a generated address is used.
- *Auditability*. Any transaction has to refer to some previous unspent transactions (unconsumed transactions). Once a transaction is recorded into the blockchain, the state of those referred unspent transactions switches from unspent to spent (consumed transaction). So transactions could be easily verified and tracked.

Blockchain technology can provide decentralized, secure, and traceable storage, attracting massive industry investment. There are currently several blockchain applications that span a vast range of industries, including IoT (Novo 2018), security (Taylor et al. 2019) (Al-Quraan et al. 2021), data privacy (Xiao et al. 2020), supply chain and goods tracing (Gayialis, Kechagias, Konstantakopoulos, et al. 2021) (Gayialis, Kechagias, Konstantakopoulos, et al. 2021), the energy sector (Qiang Wang and Su 2020), product counterfeiting (Gayialis, Kechagias, Papadopoulos, et al. 2019) and real estate (Ahmad et al. 2021) (Mehendale, Masurekar, and Patil 2019) (Ullah and Al-Turjman 2021).

2.3.2 Tangle

Tangle was created by the IOTA Foundation, a non-profit foundation incorporated and registered in Germany (Frankenfield 2021) and it was announced as the successor of Blockchain technology (Popov 2018). It is a DL designed for IoT with high scalability, zero cost, low energy consumption, and secure data transmission features (Bhandary, Parmar, and Ambawade 2020). Similar to Blockchain technology, IOTA has decentralized and tamper-proof features but doesn't work with blocks.

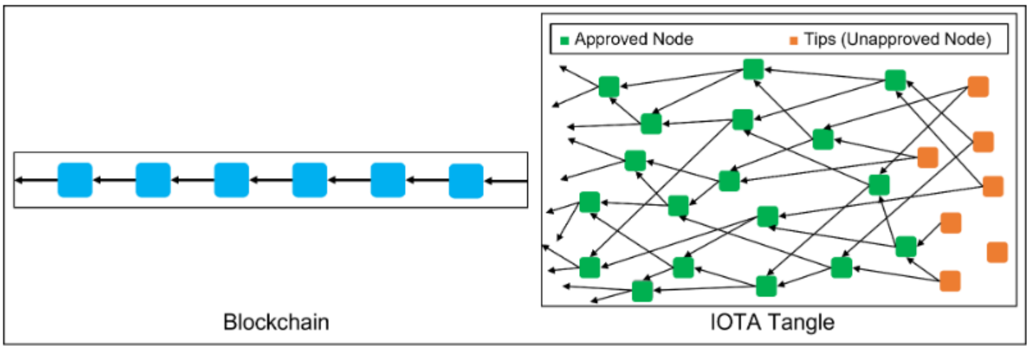

Figure 2.3: Blockchain and Tangle structure comparison (Zhang et al. 2022)

Figure 2.3 shows a comparison between the Blockchain and Tangle structure. Tangle uses the form of a Directed Acyclic Graph (DAG) for storing transactions (Shabandri and Maheshwari 2019), where the nodes represent transactions, and the edges indicate the direction between the transactions (Burkhardt, Werling, and Lasi 2018).

A DAG is defined by three traits:

- **Graph** - Composed by a series of vertexes that are connected by edges.
- **Directed** - Those edges are connected in only one way.
- **Acylic** - Impossible to traverse the graph when starting at any point.

All the nodes on the network validate transactions on the ledger. To issue a transaction, nodes must work to approve at least two randomly (assigned by an algorithm) previous transactions on the ledger in order to contribute to the network's security. Nodes that don't have two or more incoming edges are unconfirmed. After being validated, the weight of the transaction increases. The weight of a transaction works as a trust/importance indicator. The weight of a transaction is proportional to the amount of work that the issuing node invested into it. The more a node validates, the more its transactions become valid (higher initial weight) on the distributed ledger database. So, if a node has a long branch of previously validated transactions, its issued transactions will carry the most weight in the ledger (Popov 2018). However, an algorithm will randomly select the previous two transactions for each member to validate.

This approach is currently being implemented as a cryptocurrency called IOTA, which was designed specifically for the IoT industry (Dyer 2017). This is a new form of consensus in order to achieve greater scalability (Popov 2018). For every transaction that is added to the tangle, two others are confirmed. This means that the network doesn't slow when there are a lot of transactions, in fact, it speeds up.

2.3.3 Hashgraph

Hashgraph is another distributed ledger technology patented in 2016 by the CTO and the co-founder of Swirlds, Leemon Baird. One of the particularities of Hashgraph is the fact that it only uses members that are approved and authorized by the system from the start-up. Due to this it is hard to interfere with the system and it results in very fast transactions, Swirlds suggests that it can surpass 200,000 per second (D. L. Baird, Harmon, and Madsen 2020). It uses a gossip protocol that works in the following manner: Every node in Hashgraph can spread signed information (called events) on newly-created transactions and transactions received from others, to its randomly chosen neighbors. These neighbors will aggregate received events with information received from other nodes into a new event, and then send it on to other randomly chosen neighbors. This process continues until all the nodes are aware of the information created or received at the beginning (Hoxha 2018). In such an event, every participant compresses the event in a memory data structure (tuple) consisting of the hashes of two-parent events, a list of transactions, a timestamp, and a signature for the rest of the tuple (L. Baird and Luykx 2020). The event's history can be represented by a graph, presented in figure 2.4, where each member is one column of vertices (L. Baird 2016).

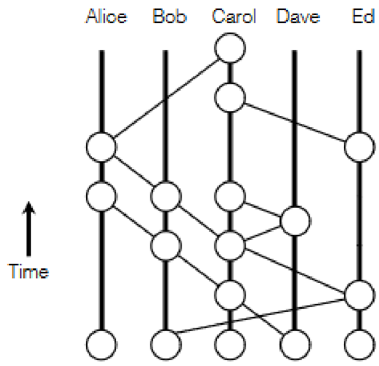

Figure 2.4: Hashgraph illustration example (L. Baird 2016)

The vertices in each column represent an event. For example, in Alice's column, the top event represents the gossip sync that was conducted by Bob, in which he sent Alice all the information that he knew (L. Baird 2016). In a typical gossip protocol, a diagram like this is merely used to discuss the protocol - there is no actual graph stored in memory anywhere. Each event is stored in-memory data structure as a sequence of bytes and signed by its creator (L. Baird 2016).

Due to this success in the corporate sector, Swirlds has now launched the "Hedera Hashgraph Platform" with aims to drive forward Swirlds' patented Hashgraph technology for the development of a public Hashgraph network (wpadmin 2017).

Hedera is governed by the Hedera Governing Council: An expert council consisting of 39 leading global organizations (Google, IBM, Boeing, and Ubisoft are among the players). The Governing Council is completely decentralized — every member has an equal vote over software upgrades, network pricing, treasury management, and more. Governing Council members are term-limited and do not receive any profits from Hedera (*What is Hedera?* 2023). Recently Hedera announced that the code was available in "Open Review", this means that the code is available only for reviewing, compiling, and testing, but not for any other use. This limitation forbids the fork of the source to create an alternative project (Hedera 2020).

2.3.4 Summary

The inherent properties of DLT such as resiliency, integrity, anonymity, decentralization, and autonomous control have fostered the early adoption of this technology in almost every application domain. However, some technologies have certain characteristics that make them a better fit for the problem that this dissertation aims to study. This section presents a summary of the DTL mentioned above.

The comparison of the functional data structures is for ledger maintenance, transaction validation, ledger size, scalability, and popularity. Table 2.1 exposes the different properties of each technology.

	Tangle	**Blockchain**	**Hashgraph**
Data Structure	Tangle structure stores its transactions in nodes, where each node holds a single transaction	Data is structured in blocks in the form of transactions that are validated by miners in the ecosystem	Data is stored in a graphlike structure
Transaction validation	All members in the network verify transactions which represents an increase in the number of members, resulting in fast validation.	Periodically batch up transactions needing confirmation — into a block — and confirm them in one go using a consensus mechanism	The gossip protocol and virtual voting ensure that the majority validates transactions.
Transactions per second	Using a DAG data structure ensures high scalability and high TPS	Limited scalability and TPS	The unique consensus protocol reduces the computational load and provides higher scalability and higher TPS
Patented	No	No	Yes

Limitations	Low transactions volume make Tangle more vulnerable to attacks	Low transaction speed and High power consumption	The technology has only been deployed in a private network although, the real potential will only be known once it is released in a public network (Takyar 2018). Any new invite to the ledger will rely on and go through Swirlds

Table 2.1: DLT Comparison

For the context of this dissertation, an essential requirement that the DLT should have is security. Real estate data is susceptible, which requires the minimization of risks by using mature and accessible distributed ledger technology. Data must be resistant to attacks even with low transaction volume. In terms of transaction volume, a DAG implementation when faced with low transaction volume can be skeptical to attacks that can lead to security breaches resulting in the ledge's instability. Since this project will not deal with a massive amount of data and with high operational demand, the complexity of a DAG implementation does not justify the business needs. Real estate data is composed of private and sensitive information, so the speed and transaction fees can become less critical while security becomes the primary concern. To conclude, despite its inefficiency (mostly in public platforms), blockchain technology continues to be the most tested, used, and versatile DLT available today and is the chosen DLT for this dissertation proof of concept.

2.4 Blockchain in-depth analysis

Blockchain can be pictured as a layer of a distributed peer-to-peer network being executed on top of the internet. The blockchain is analogous to SMTP, HTTP, or FTP running on top of TCP/IP (Bashir 2020).

Figure 2.5 shows an overview of blockchain technology architecture. The blockchain can be divided into four layers (Ismail and Materwala 2019): infrastructure, platform, distributed computing, and application.

The infrastructure layer consists of all the hardware components required to run the blockchain, such as nodes, storage, and network facilities. The nodes are the network participants. A typical blockchain network has three different types of nodes: a simple node (also referred to as a light node), a full node, and a mining node: A simple node in the network can just send and receive transactions and does not store a copy of the ledger, nor validate a transaction, whereas a full node does. A Mining node (also referred to as a block generator) is a full node with the capability of mining, i.e., the process of generating a new block.

The storage component stores the ledger of the transaction records. The platform layer facilitates remote procedure calls (RPC), web application programming interface (API), and REST API's for the communication between the network participants.

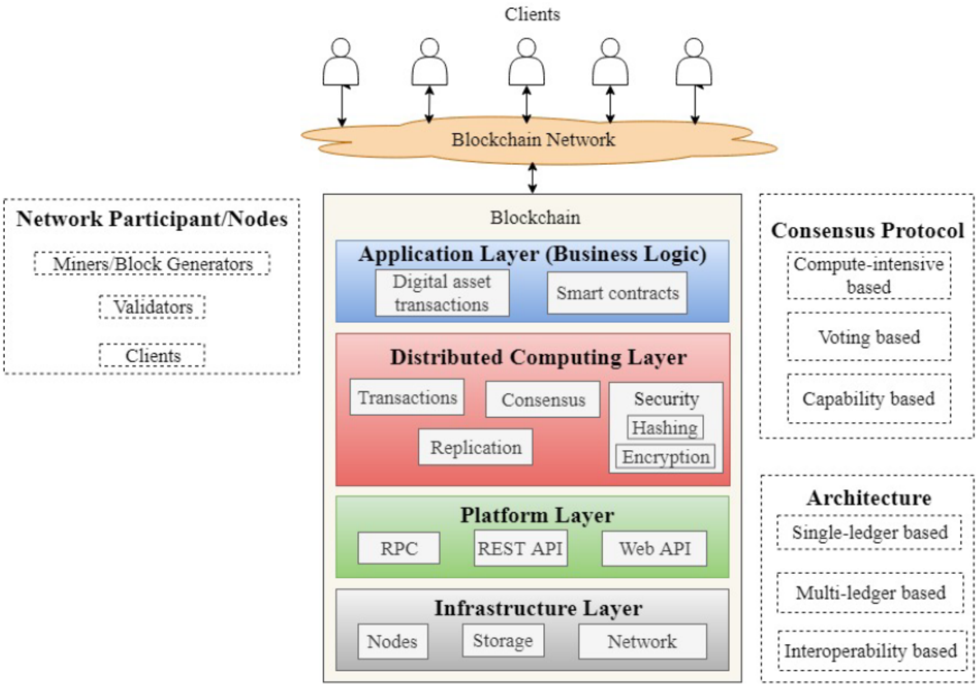

Figure 2.5: Blockchain Overview (Ismail and Materwala 2019)

Blockchain technology can be implemented in various economic sectors, particularly in the finance sector, where improvements in the performance of financial transactions can be significantly beneficial (Bashir 2020).

The proliferation of digital everything as both a means of exchange and a store of value has expanded significantly, with a seismic shift impacting the global financial services industry (FSI) in particular that has recently been further challenged by new business models around digital assets. Deloitte conducted a Global Blockchain survey in 2021 (Budman et al. 2021), from a sample of 1,280 senior executives and practitioners in 10 locations around the globe, as a research vehicle to gain insights into overall attitudes and investments in blockchain and digital assets. The vast majority of leaders in this survey (97% of FSI Pioneers) see blockchain and digital assets as another way to gain a competitive advantage. Other conclusions of the survey were:

- In terms of blockchain use cases: Security information represents the most-cited blockchain use case, followed by Digital Currency, Asset Tracking & Management, and Digital Identification.
- For the most-valued digital identity opportunities: Spotting Fraudulent invoices was the top pick followed by Signing Contracts and Enabling Financial Inclusion (access to the unbanked)

2.4.1 Is Blockchain trust-free?

Trust, it has been said, is "the bond of society"(Yeo 2013) and society would hardly be able to function without it. According to Duranti and Rogers (Duranti and Rogers 2014), "Trust has been defined in many ways. In business, trust involves the confidence of one party in another, based on an alignment of value systems with respect to specific benefits in a relationship of equals. In jurisprudence, trust is usually described as a relationship of vulnerability, dependence, and reliance in which we participate voluntarily. In substance, trust means having the confidence to act without the full knowledge needed to act. Several authors indicate "trust-free" as an inherent property of the blockchain, either by drawing upon the blockchain's technological foundations or by citing related work. Thus, according to this perspective, the blockchain is something that is trust-free. For example, Notheisen et al.(Notheisen, Cholewa, and Shanmugam 2017) already indicated in the title of their paper that trading real-world assets on a blockchain is "an application of trust-free transaction systems". To state another example, Schweizer et al. (Fridgen et al. 2017) wrote that "blockchain systems are generally considered to operate as closed trust-free ecosystems"

In sharp contrast to this perspective, other statements in the literature suggest that the blockchain is not trust-free. For example, Siira et al. (Siira et al. 2017) wrote that "utilizing blockchain technology ... would offer ... possibly more trustworthiness". As another example, Cao et al. Cao et al. 2017) wrote that "blockchain technology has a great influence on reducing the risk of trust".

According to Auinger (Auinger and Riedl 2018), claiming that the blockchain is trust-free is "a massive exaggeration... Trust only shifts from specific market players in the blockchain ecosystem to others. However, this does not mean that trust issues have changed fundamentally. Rather, traditional determinants of trustworthiness (i.e., ability, benevolence, integrity), along with known mechanisms to establish trust in online settings (e.g., third-party institutional mechanisms), will remain critical in blockchain settings."

In practice, blockchain technology really does not obviate the need for trust. Instead, it offers a new way to substitute the information one does not have from other sources in order to place confidence in something or someone and, by extension, take action on the basis of having that trust. It purportedly serves to replace more traditional, and often very inefficient or flawed means of obtaining this information and establishing trust with a new, more efficient source of information as a basis for trust.

2.4.2 Taxonomy of Blockchain Systems

Currently, blockchain systems can be implemented in various types as presented in figure 2.6, mostly depending on the business context:

- Public - This type of blockchain is open to everyone who intends to become a node in the system. Each user maintains a copy of the ledger and uses a consensus mechanism to decide its state of it. Popular examples are Bitcoin, Ethereum, and Litecoin.
- Private - As the name implies, only a group of users or an organization can access and share the ledger among themselves (Polge, Robert, and Le Traon 2021). Hyperledger Fabric, R3 Corda, and Ethereum Enterprise are examples of private Blockchains.
- Consortium - Also known as a federated blockchain, this approach has all the same benefits as a private blockchain but unlike private blockchains, consortium or federated blockchains are usually not owned and used by one sole group or organization. Rather,

multiple organizations can exist on a single federated blockchain, allowing them to share data across the network privately and securely. Hyperledger Fabric is one of the platforms to implement this type of blockchain (Zeng et al. 2019).

- Hybrid - The hybrid blockchain is a combination of public and private entities. The best way to describe it is using a public blockchain where a private network is hosted. This means that there is restricted participation that is controlled through the private blockchain itself. Hybrid is best suited for projects that can neither go private nor public and have a lack of trust. The supply chain is a great example. It is also effective in banking, finance, IoT, and others (Geroni 2021).

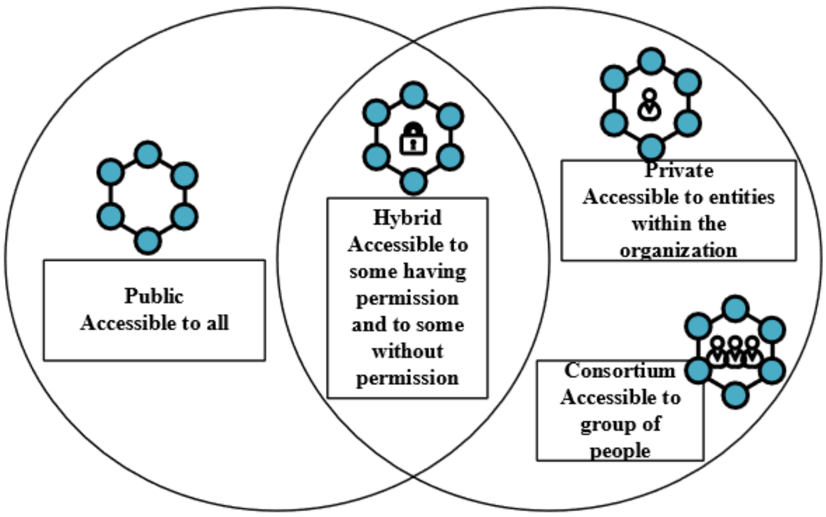

Figure 2.6: Types of Blockchain (Ismail and Materwala 2019)

In the context of this project, in order to develop a Proof of Concept of an MLS, a private blockchain system would fit better as only selected members of one organization will have access to make transactions and, with that, modify the ledger.

2.4.3 Consensus mechanisms

There is a need to ensure that all nodes are consistent with each other in a distributed system. In a blockchain, there is no central authority that ensures that different nodes have the same ledger and are consistent. Therefore, it is necessary to implement a consensus approach. This section presents some of the most known consensus mechanisms used in the blockchain environment. All the consensus mechanisms fall into two main categories (Zheng, S. Xie, H. Dai, et al. 2017):

- Proof-based or leader-election lottery-based consensus is a type of agreement where a leader is elected randomly (using an algorithm) and proposes the final value. This type of category refers to the public types of consensus mechanisms. Some of the above mention consensus mechanisms that fit into this category are, for example, Proof of Work and Proof of Stake.

- Byzantine Fault Tolerant or BFT-based, on the other hand, is a traditional approach based on rounds of votes. This type of consensus requires that every node be known to the network. An example of this type of category is, for instance, pBFT.

Proof of Work (PoW)

In PoW, only one node can be selected to record a block. This selection cannot be random because it would make the system vulnerable to attacks. Therefore, selecting a trusty node is necessary for some computer calculations. In PoW, nodes compete with each other to search for a nonce, by sheer brute-force use of processing power, until the resulting block hash follows a particular pattern of a certain number of zeros (Zheng, S. Xie, H. Dai, et al. 2017). In this consensus, the selection of the node is purely based on the proportion of their computing capacity. When one of the nodes reaches the target value, it broadcasts the block to other nodes that must confirm the correctness of the hash value. If the block is valid, the miners (nodes that calculate the hash values) will append this new block to their blockchains. This consensus mechanism is used by Bitcoin (Nakamoto 2008) and is designed to specifically address Byzantine faults (Lai and LEE Kuo Chuen 2018).

Proof of Stake

Proof of Stake was first introduced by Peercoin, in 2012 (Peercoin, 2012) and is an energy-saving alternative to PoW. The selection process, in this consensus, takes into consideration the proportion of tokens a node already owns. It is believed that the more tokens a node possess, the less the probability of this node attacking the network (Zheng, S. Xie, H. Dai, et al. 2017). To lunch a successful attack, the node must acquire many tokens which makes the attack economically unsustainable (Lai and LEE Kuo Chuen 2018).

Centralizing the power of the network in the richest nodes makes PoS more effective than PoW but, since the mining cost of PoS is low (practically zero), the system is more likely to suffer an attack. Many blockchains are adopting, at first PoW, and then gradually move to a PoS (Zheng, S. Xie, H. Dai, et al. 2017). On September 15, 2022, Ethereum's transition from PoS was completed, officially deprecating the former proof-of-work consensus mechanism and reducing energy consumption by 99.95. This is also known as The Merge (Calma 2022).

Practical Byzantine Fault Tolerance

In 1999, Castro and Liskov published Practical Byzantine Fault Tolerance, or pBFT, which provided a BFT for practical use (Castro and Liskov 2019). It describes the first state-machine replication protocol that correctly endures Byzantine faults in asynchronous networks. Their algorithm "can be used to implement any deterministic replicated service with a state and some operations" (Castro and Liskov 2019). pBFT guarantees liveness – progress will be made within the system – and safety – the chance that something negative will happen in the system – being able to handle up to 33% of nodes being faulty (Lai and LEE Kuo Chuen 2018).

Raft

Raft is another fault-tolerant consensus protocol created by Diego Ongaro and John Ousterhout in 2014 (Ongaro and Ousterhout 2014). It aims to be a simplified version of Paxos as it was designed to be easier to understand and implement. Raft consensus protocol has had a massive adoption, especially in the open-source community. Both protocols are leader election-based consensus mechanisms, meaning that nodes are competing in a leader-election lottery, and the node that wins proposes a final value Bashir 2020. The critical difference in the aforementioned protocols is that Raft selects the leaders among the new nodes while Paxos selects the leader among all nodes (Lai and LEE Kuo Chuen 2018).

Chapter 3

State of the Art

In recent years, distributed ledger technology and blockchain have emerged as powerful tools for creating decentralized, secure, and transparent systems for recording and tracking transactions. These technologies have the potential to revolutionize a wide range of industries, from finance and banking to supply chain management and digital identity verification. This state-of-the-art chapter will provide an overview of the current state of distributed ledger and blockchain technology, including key concepts, key players, and notable use cases. It will also present a review of the major technologies for the use of a blockchain-based system. Overall, this chapter will aim to provide a comprehensive overview of the current state of the art in distributed ledger and blockchain applications and their potential to shape the future of the real estate industry.

3.1 Generations of blockchain technology

In 1982, Chaum was the first known person to propose a blockchain-like protocol in his Ph.D. thesis (Chaum 1982). In 1991, Haber and Stornetta described a secured chain of blocks cryptographically (Haber and W. S. Stornetta 1991). In 1993, Bayer et al. incorporated Merkle trees into the design (Bayer, Haber, and W. Stornetta 1999). We can gather these events as a "prelude" to blockchain technology as we know it today.

Blockchain technology can be viewed as having gone through three stages and is well into the third phase of development (Franks 2020):

Blockchain 1.0

Blockchain 1.0 introduced financial transactions and payments via cryptocurrency. In 1998, a decentralized digital currency ("bit gold") mechanism was designed by Szabo (Sharma 2021). However, Satoshi Nakamoto's Bitcoin was the first "mass-used" transaction-focused blockchain system (Nakamoto 2008). Since then, over 20,000 different cryptocurrencies have been introduced. Those financial transactions taking place in the first phase (and continuing today) involve nodes on a distributed network. This absence of a third party in the financial transaction process (i.e. a bank) by being replaced for the "trust" provided by the secure distributed ledgers (similar to those used by cryptocurrencies) is an intrinsic part of the Decentralized finance (DeFi) concept (Sharma 2022).

Blockchain 2.0

Blockchain 2.0 emerged when Ethereum introduced distributed applications and smart contracts that reside atop a blockchain. Ethereum also introduced the Solidity Programming language, a contract-based, high-level language to code smart contracts (Moura 2020). Although Ethereum was a pioneer in the "programmable blockchain", other platforms like Hyperledger Fabric and Corda allow the development of these types of contracts.

The basic idea of smart contracts is that many kinds of contractual clauses (such as liens, bonding, delineation of property rights, etc.) can be embedded in the hardware and software we deal with, in such a way as to make a breach of contract expensive for the breacher (Szabo 1996). Smart contracts perform transactions (agreements), between two or more parties, when a specific set of rules are followed, such as when both parties sign an agreement, or it can be triggered by external data, such as the passing of an expiration date or the attainment of a price goal. If the information necessary to trigger the contract is located outside the blockchain network (i.e. off-chain), an external actor (such as an oracle) can be used to find and verify real-world occurrences and feed the data to the blockchain network (Franks 2020).

Nick Szabo's vending machine is a canonical example (Szabo 1996). Szabo declares that inside the vending machine, there is a sort of proto-smart contract, containing a set of computer code that looks something like what is presented in the listing 3.1. Two conditions are required in order to get a Coca-Cola can:

1. The button to be pressed needs to contain the label "Coca Cola"
2. The amount of money inserted in the vending machine must equal or higher than 1.75

```
if button_pressed == "Coca Cola" and money_inserted >= 1.75:
    release("Coca Cola")
    return_change(money_inserted - 1.75)

else if button_pressed == "Aquafina Water" and money_inserted >= 1.25:
    release("Aquafina Water")
    return_change(money_inserted - 1.25)
```

Listing 3.1: Smart contract logical example by Nick Szabo (Szabo 1996)

Smart contracts can be developed or combined to build *dApps*, which are applications that run on a peer-to-peer network. Because smart contracts are self-executing, dApps, as configurations of smart contracts, are as well — and thus are not subject to the control of any centralized authority. Instead, dApps rely on their underlying blockchain for any coordination of their operations (Brummer 2022).

According to Raval (Raval 2016), dApps are characterized by four properties:

- Open Source: Due to the trusted nature of blockchain, dApps need to make their codes open source, so that audits from third parties become possible.
- Internal Cryptocurrency Support: Internal currency is the vehicle that runs the ecosystem for a particular dApp. With tokens, it is feasible for a dApp to quantify all credits and transactions among participants of the system, including content providers and consumers.
- Decentralized Consensus: The consensus among decentralized nodes is the foundation of transparency.
- No Central Point of Failure: A fully decentralized system should have no central point of failure since all components of the applications will be hosted and executed in the blockchain.

Some examples of dApps are (O'Neill, Hussey, and Chipolina 2022) :

- Uniswap - A decentralized exchange that enables users to swap tokens peer-to-peer rather than through a centralized intermediary.
- MakerDAO - A smart contract that enables users to interact with the DAI stablecoin system.
- Gods Unchained - An NFT-powered card game.

By the first quarter of 2022, there were almost 2.4 million daily active users of dApps (O'Neill, Hussey, and Chipolina 2022). However, dApps can be vulnerable to hacks. In the first quarter of 2022 alone, $1.2 billion was stolen in hacks and exploits, according to DappRadar. In August 2021, Poly Network was exploited for $611 million (Mayer 2021) and in March 2022 Axie Infinity's play-to-earn game Ronin bridge was hacked for $552 million (Jha 2022). Before dApps reach the mainstream, developers and the networks on which they build dApps have a long list of challenges to work through, including scalability, security, and UX.

These contracts also enable new governance models like a Decentralised Autonomous Organization (DAO). Vitalik Buterin, the Ethereum co-founder, described DAOs as "an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do." (Buterin 2014). The difference between a DAO and a typical organization is that there is no need for any formal management structure, instead, it runs on rules encoded on smart contracts (Zapotochnyi 2022). The rules of the DAO are established by a core team of community members through the use of smart contracts. These smart contracts lay out the foundational framework by which the DAO is to operate (Shuttleworth 2021). The goal is to create a way for organizational decisions to be made with regard to the needs and desires of its shareholders, without being slowed down by the diversified opinions and interests associated with people working within a traditional framework (Zapotochnyi 2022). In the first attempt at recreating a fully-functioning decentralized organization, a project with a direct self-explanatory name "The DAO", ended in controversy when a hacker took advantage of a loophole in the smart contract and transferred up to $50 million worth of Ether into his wallet (DuPont 2017). The majority decision for a solution was that Ethereum needed to create a fork, or stop the blockchain entirely and create something new from scratch. This "something new" is what now is seen as Ethereum (ETH). Ethereum Classic (ETC) is, as the name would suggest, the first Ethereum still using the original blockchain. The actual Ethereum chain that forked was able to get back the $50 million that was hacked (Moskov 2020). UkraineDAO is a DAO organizing and funding efforts related to helping Ukrainian victims of war and supporting the Ukrainian defense effort. In about a month's time, the Ukraine DAO raised over $8 million for organizations and people aiding Ukrainians amid the war (Locke 2023). Among the most notorious supporters of this project is Vitalik Buterin, the Russian Ethereum co-founder (Locke 2023).

Directly connected to smart contracts is the concept of non-fungible tokens (NFT). An NFT can be said to be a cryptocurrency but it differs from classical cryptocurrencies such as Bitcoin in their intrinsic features. Bitcoin is a standard coin in which all the coins are equivalent and indistinguishable. In contrast, NFT is unique and cannot be exchanged like-for-like (equivalently, non-fungible), making it suitable for identifying something or someone in a unique way (Qin Wang et al. 2021). An NFT can be used to represent ownership of unique items: art, collectibles, and even real estate can be tokenized. Tokenization is a method that converts ownership rights to an asset into digital tokens that can be bought, sold, and traded on blockchains (Sazandrishvili 2020). Propy, the first blockchain startup to smooth real-world estate sales, by introducing the concept of smart contracts, was responsible for the auctioning of a real apartment as an NFT. The auction of the NFT was attached to a modern, brand-new, one-bedroom apartment in Kiev, Ukraine. Propy previously made history with this property by making it the first-ever blockchain-based real estate sale (Masse 2017). The winning bidder was granted the NFT which includes access to the ownership transfer paperwork, a digital artwork NFT by a popular Kiev graffiti artist, and the apartment pictures. But obviously, the apartment is the main asset acquired (Butcher 2021).

Blockchain 3.0

For the foreseeable future, not all data will reside on a blockchain. If Blockchain technology is going to be incorporated into the operations of most industries, extensibility beyond the blockchain is essential. Blockchain 3.0 recognizes the necessity of interoperability between the blockchain network and other systems and services. This generation of blockchain brings us blockchain as a service (BaaS), where established vendors leverage their resources to provide customer services. The Blockchain as a Service or "Blockchain-as-a-service" (BaaS) is a union of Blockchain and Cloud computing (Melo et al. 2018).Customers can take advantage of cloud-based solutions to build, host, and use their own blockchain apps, smart contracts, and functions on the blockchain. The cloud-based service provider manages all the necessary tasks and activities to keep the infrastructure agile and operational Franks 2020.

Some of the largest Blockchain-as-a-Service (BaaS) providers are:

- Amazon - Launched in 2018, Amazon Managed Blockchain-AWS is the most popular BaaS platform that makes it simple to manage scalable blockchain networks using Ethereum and Hyperledger Fabric frameworks. This provider automatically scales to meet the demands of thousands of applications running millions of transactions and eliminates the expenses required in the creation of the network.
- Corda - R3's Corda advertises itself as "the only multi-party app development platform purpose-built for highly regulated industries" and focuses primarily on banking, capital markets, trade finance, and insurance, industries where top-tier compliance and privacy practices are a must. R3 also includes Conclave, their confidential computing platform for application development.
- Kaleido - Kaleido is one of the most comprehensive blockchain-as-a-service solutions on the market that considers cross-cloud and hybrid deployments from the outset. While many other BaaS solutions focus on quick-start scripts, templates, and other basic needs at the launch/implementation phase of blockchain development, this private blockchain network moves past the basics with robust native and API integration offerings and advanced digital governance technologies. Kaleido is also known for its user-friendly interface and codeless development options, originally only offered in Ethereum but now also offered in Corda and Hyperledger Fabric protocols.

3.2 Real-World Applications of Blockchain in Real Estate

Several governmental and enterprise applications of blockchain technology, in different phases of exploration, have been taken up from around the world. These projects are discussed below:

3.2.1 Enterprise solutions

- Propy: A global decentralized property store on the Ethereum blockchain where the focus is to close a traditional real estate deal entirely online. For that, purchase agreements are signed with DocuSign, which is a digital transaction management platform that allows individuals and organizations to manage electronic signatures and other digital document processes (Norton 2021). Most of the transactions are done in dollars in Propy, an attempt to make it more viable to crypto skeptics, as it's able to process wire transfers via integration with a money transmitter connected to 70 banks. It is also an enterprise play going on here as well, as it can provide the back-office system to real estate enterprises with real-time transaction reports and automated compliance. Propy claims its platform saves 10 hours of paperwork, per transaction (Butcher 2020). Natalia Karayaneva, one of Propy's founders, said in a statement that the "platform offers a terminal to observe transactions in real-time, making the process transparent for real estate executives, title companies, homebuilders, buyers, and REITs." (Butcher 2020). Propy is not a Multiple Listing Service as its main purpose is to manage transactions for the properties listed in multiple MLS.
- ShelterZoom: ShelterZoom launched the platform 1REport, an easy-to-use platform that offers potential investors a way to enter the blockchain property market from the desktop/mobile platform (ShelterZoom 2021). Transactions are founded by smart contracts in the Ethereum network, attracting users from over 22 countries. Another key feature of ShelterZoom's platform is its tokenization capability. The platform allows properties to be tokenized, which means that they can be divided into smaller portions and sold as fractional ownership (Gupta et al. 2020). This can make real estate investments more accessible to a wider range of investors, as they can participate in the market with smaller amounts of capital.

3.2.2 Countries/Cities

- Republic of Georgia - The government of Georgia initiated a project for the implementation of Blockchain technology in land titling as an association between NAPR (National Agency of the Public Registry) and Bitfury. The implementation strengthens the system by incorporating trust into the system. It also helps in the prevention of fraud by verifiability of data (Shang and Price 2018). The existence of supportive data protection laws in the country is an opportunity for the technology Lazuashvili 2019. The weakness of the system can be defined as insufficient public awareness of the technology. The implementation brings along an opportunity to attract foreign investors due to the increased ease of doing business (Rodima-Taylor 2021). For this project to succeed an important factor is the education of the public about the technology, which can pose a threat to the same if it is not taken care of.

- Sweden- The possibility of application of blockchain technology for real estate in Sweden was initially explored by Kairos Future (The strategy consultant of Lantmäteriet), Telia (the telecom company), and ChromaWay (a blockchain start-up) in 2016 (Dreyfuss-Chavez 2016). The project was driven by the incompetency of the existing system of land registration including time-consuming processes, delays in transfer, manual processes of verification, and vulnerability to error among others (McMurren, Young, and Verhulst 2018). The implementation of blockchain technology will strengthen the system by making it secure, efficient, trusted, less vulnerable to error & fraud, transparent and fast. The validity of digital signatures is still uncertain for real estate contracts which depicts a weakness of the model. Since the project was successful in the trial phases, there lies a vast opportunity to scale it up to process real land transfers. With the increased scale of operations, the need for infrastructures like servers, storage, and nodes for blockchain verification will increase thus posing a threat to the system. Another challenge for the system will be the inclusion of various parties to the system like realtors, buyers, sellers, etc. which are still not there in the system.
- Dubai- Dubai land department implemented blockchain technology for registration of property, buying, selling and mortgages, etc, as a part of the Smart Dubai Office initiative (Bishr 2019). The strengths of this project are the key features of blockchain technology including immutability of records, faster processes, cost savings, and security and verifiability of transactions. Since it is a relatively new technology, lack of awareness among the public and limited availability of skilled human resources can be termed as its weakness. The implementation of this technology has significantly improved Dubai's ranking in the Ease of Doing Business Index thus making it a more attractive destination for investments which can be termed as a great opportunity. Apart from this, the implementation provides the government an upper hand in its aim of becoming the world leader in blockchain intellectual capital and skill development Farghaly et al. 2019. The absence of regulatory frameworks and data protection laws could pose a threat to this project.
- Honduras - The government of Honduras in collaboration with two companies namely Factom and Epigraph agreed to build a permanent and secure land title record system. The need for this project arose out of widespread corruption, time-consuming registration processes, and land title frauds happening in Honduras. The main strengths of the project include increased transparency in a transaction, precise, verifiable transactions, and an immutable register resulting in a lower number of disputes in the future. These strengths result in creating opportunities like opening economic growth opportunities by lowering the cost of borrowing. Also, a lesser number of property disputes in the future will provide respite to the judicial system. The system also has some weaknesses like high costs of development, implementation, and setup of hardware. The threats associated with the project include long-term investments and a lack of laws available to deal with (*Honduras to build land title registry using bitcoin technology* 2015;Collindres, Regan, and Panting 2016). Another innovation relating blockchain and Honduras is the fact that a private charter city and special economic zone named Próspera announced that "Bitcoin and other cryptocurrencies effectively operate as legal tender within its jurisdiction". (Prospera 2022).
- Ghana- In association with Bitland (A USA-based platform), Blockchain technology was introduced in the land registry system in Ghana Kshetri and Voas 2018. The

trust, immutability, and verifiability properties of blockchain technology are the major strengths of the project (Mintah et al. 2020). The introduction of blockchain technology will provide various opportunities enabling the use of land capital and allowing mortgages to support development processes (EconoTimes 2016). The implementation of blockchain technology in land title registration is not in coherence with the existing laws and regulations of the country, which is a weakness of the system. This needs to be addressed before the complete implementation of the system. The digital records are placed on the ledgers having control outside the jurisdiction of the Government of Ghana which can pose a threat to the immutability and privacy of records (Mintah et al. 2020).

3.3 Blockchain Technology Platforms

Blockchain technology has revolutionized the way we store and transfer data, making it more secure, transparent, and decentralized. With the increasing popularity of this innovative technology, the number of blockchain platforms available in the market has also grown significantly. Selecting the right platform that fits the needs of a project can be a challenging task, given the wide range of options available. However, due to the problem deriving from an enterprise solution, a permissioned blockchain network offers more guarantees in terms of security and trustworthiness among nodes. In this section, some of the most widely used and well-known permissioned blockchains in the enterprise space are presented (Polge, Robert, and Le Traon 2021).

3.3.1 Hyperledger Fabric

The Linux Foundation founded the Hyperledger project in 2015 to advance cross-industry blockchain technologies. Hyperledger Fabric is one of the blockchain projects within Hyperledger such as Hyperledger Sawtooth, Indy, Caliper, and Ursa (Rofiq and Rafid 2021). It is a decentralized operating system for permissioned blockchains that can execute distributed applications (Dapps) written in general-purpose programming languages such as Go, Java, or JavaScript (node.js) (Androulaki et al. 2018).

The Fabric network is composed of nodes whose identities are given by a Membership Service Provider (MSP). These nodes can be (Polge, Robert, and Le Traon 2021):

1. **Clients** that propose transactions to execute and broadcast them for ordering
2. **Peers** that maintain the state of the ledger
3. **Ordering service** nodes that establish the order of all the transactions. This type of node does not participate in the execution of the validation processes.

Hyperledger Fabric also implements smart contracts, called chaincode, for the implementation of the application business logic. This platform natively implements Solo, a voting-based consensus protocol consisting of *endorsing nodes* executing the transactions and validating them in compliance with the *endorsement policy*. However, Fabric allows several pluggable options, ledger data can be stored in multiple formats, and other consensus protocols such as Practical Byzantine Fault Tolerance (PBFT), Raft, or Kafka (to be able to use several ordering nodes) can be plugged in.

Privacy is a reason why companies look at private blockchains in the first place. In order to control the privacy of the participants in a private/permissioned Hyperledger Fabric network, a trusted Membership Service Provider (MSP) is required. The Membership Service Provider maintains the identities of all nodes in the system (clients, peers, and ordering service nodes) and it is responsible for issuing node credentials that are used for authentication. (Androulaki et al. 2018).

This platform also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions. This feature is particularly crucial for networks involving competitors who prefer to restrict access to specific transactions, ensuring confidentiality among participants. If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel. Finally, Hyperledger Fabric has no underlying cryptocurrency.

3.3.2 R3 Corda

R3 Corda is an open-source blockchain permissioned platform developed by R3. It runs on the JVM and there's no blockchain, unlike other ledger platforms. Instead, each node only sees and stores a subset of transactions on a ledger. If some transaction is processed between X and Y, this transaction (including related all states) is stored in X's and Y's ledger, but it's not stored in Z's ledger (Hearn and Brown 2019). This design allows Corda to process multiple transactions simultaneously, while not putting privacy requirements at risk.

A Corda network consists of the following components (Hearn and Brown 2019):

- **Nodes**, representing people and businesses, are arranged in an authenticated peer-to-peer network. All communication is direct but no *gossip* protocol is used. Each node maintains a ledger and a vault for persistence.
- An **Identity service** which assigns an identity to a node.
- A **Network map service** that publishes information about how to connect to nodes on the network, the identity of the nodes is represented by an IP address.
- A **notary service** is responsible for providing uniqueness consensus. These are special nodes that will ensure that input is consumed one time. It prevents double-spending.
- Zero or more oracle services. An oracle is a service done by nodes by which off-ledger data is introduced onto the ledger.

Corda follows the "Know Your Customer" principle, each node has to prove its identity to be authorized to join the network (Polge, Robert, and Le Traon 2021). The Doorman is the node in charge of validating the identities and distributing the certificates. The network is also composed of one or many Notary nodes, their role is to validate the uniqueness and the sequencing of the transactions without global broadcasting. Two types of consensus have to be reached in Corda: validity and uniqueness. Validity is checked by each signer before signing the transaction, and uniqueness is checked by the Notaries. Smart contracts written in JVM languages are supported.

3.3.3 Quorum

Quorum is a permissioned blockchain based on the Ethereum blockchain. It has been developed by J.P. Morgan for financial use cases but can be used for any type of industry (Polge, Robert, and Le Traon 2021). Compared to Ethereum it brings several changes, or in some cases, enhancements:

- Permissioning: Only authorized peers can participate in the network.
- Privacy: "Public" transactions are only visible to all the participants of the permissioned network. It is also possible to create private contracts and transactions whose payload is only visible to the participants that are specified in a parameter of the transaction.
- Alternative consensus protocols: Other protocols for consortium blockchains such as a Raft-based consensus protocol and Istanbul BFT are also available (Polge, Robert, and Le Traon 2021).
- Higher performance: A culmination of the previous points, due to the permissioned nature and therefore limited to the number of participants on the platform, a higher level of performance compared to the public Ethereum blockchain will be achieved.

3.3.4 MultiChain

Developed by Coin Sciences, it is an open-source platform that is a fork of the Bitcoin blockchain (Greenspan 2015). However, unlike Bitcoin, MultiChain is a blockchain platform optimized for enterprise use cases allowing users to configure several parameters (Polge, Robert, and Le Traon 2021):

- Manage permissions and privacy in the chain.
- The maximum block size.
- The mining process is conducted by a set of identified block validators. Also, there's only a single validator per block.
- The development of smart contracts, supporting a variety of programming languages such as Python, C#, PHP, Ruby, or JavaScript.

Chapter 4

Value Analysis

Lawrence D. Miles, a former manager of Value Services at General Electric and named "the founder of value analysis/engineering", in his book 'Techniques of Value Analysis Engineering' defined Value Analysis as "an organized creative approach which has its purpose the efficient identification of unnecessary cost i.e. cost which provides neither quality nor use nor life nor appearance nor customer features." (L.D. Miles 2015). The primary objective of value analysis is to assess how to increase the value of an item or service at the lowest cost without sacrificing quality (L.D. Miles 2015).

In this chapter, the application of various tools and techniques is employed in order to understand if the developed idea really creates value for the contextual business.

4.0.1 Innovation process

Koen described the innovation process as composed of three phases: Fuzzy Front End (FFE), New Product Development (NPD), and the Commercialization phases, as indicated in Figure 4.1 (Koen 2007).

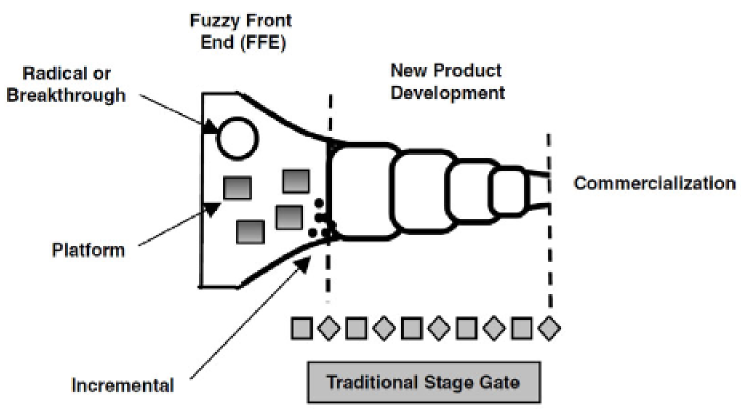

Figure 4.1: Innovation Process

- Fuzzy Front End (FFE). Provides the innovation concepts and business plans for the NPD phase. These pre-product development activities are often chaotic, unpredictable, and unstructured. The step generates innovative ideas and solutions that may be rigorously designed in future steps.
- New Product Development (NPD). Is a set of formal, typically structured activities that cover the process of bringing a new product to market availability.

- Commercialization. The process of bringing the developed products to the market.

The front end of innovation, or what is often called the Fuzzy Front End, presents one of the most significant opportunities for improving the innovation process. This phase is receiving more attention because it perceives that the New Process Development phase was not receiving high-profit ideas coming from the FFE.

The New Concept Development Model (NCD), shown in Figure 4.2, provides a common language and definition of the FFE, and consists of three essential parts (Koen 2007):

1. The inner area defines the five elements of the Front End of Innovation (FEI).
2. The engine that drives the five front-end elements and is powered by the organization's leadership and culture.
3. The influencing factors consist of Organizational Capabilities, Business Strategy, and the Outside World – distribution, channels, customers, and competitors.

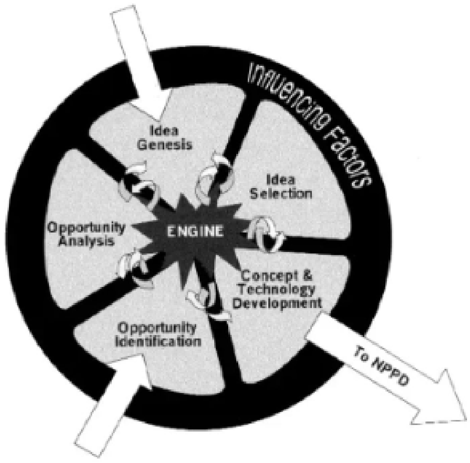

Figure 4.2: The New Concept Development Model (NCD)

The front-end elements that are a part of the New Concept Development Model are (Koen 2007):

1. Opportunity Identification. This is where the organization identifies the opportunities that might be interesting to pursue. Business and technological opportunities are considered, and resources are allocated to new areas of market growth—the goals of the business drive this element
2. Opportunity Analysis. Translating an Opportunity Identification into specific business and technology opportunities requires additional information for making early and uncertain technology and market assessments. Both competitive intelligence and trend analyses are used in this element.
3. Idea Genesis. Genesis is the birth, development, and maturation of the opportunity into a concrete idea. The idea may go through many iterations and changes as it is analyzed, studied, discussed, and developed. The goal of this element is to generate new or modified ideas for the identified opportunity.

4. Idea Selection. This element's goal is to choose which ideas achieve the most business value. It is important to note that idea selection must be less rigorous than opportunity analysis since the ideas should be allowed to grow and advance with less certainty.
5. Concept and Technology Development. The final element of the model involves the development of a business case based on estimates of market potential, customer needs, investment requirements technology unknowns, and overall project risk.

4.0.2 Opportunity Identification

As stated before, an MLS is an internal database used by real estate brokers and agents to share information about properties for sale or lease. By implementing blockchain in an MLS for the management of exclusivity contracts, the following benefits could be achieved:

- Security: Ensuring that the information in the exclusivity contract is accurate and tamper-proof. This can reduce the risk of fraud or data breaches, which can be a major concern in the real estate industry.
- Transparency: A transparent and auditable record of all details in exclusivity contracts, making it easier to track possible amendments. This can improve the accuracy and reliability of the information in the MLS, and increase confidence among buyers and sellers.
- Smart Contracts: They enable the use of smart contracts, which are self-executing contracts with the terms of the agreement written into code. This can automate the management of exclusivity contracts, for example the commissions accorded in the terms of the contract can be automatically distributed to the actors involved in the sale process or trigger a contract extension when a certain clause is met. Another situation is the verification of irregularities such as a duplicated exclusivity contract involving the same property but different agents.
- Opportunity for collaboration among different agencies - Blockchain technology serves as an impartial intermediary in real estate transactions involving multiple agencies, such as Agency X presenting an opportunity to sell a property held by Agency Y. Through its decentralized and transparent ledger, blockchain ensures fairness, transparency, and trust, benefiting both parties in this collaborative venture.

4.0.3 Opportunity Analysis

In order to ground the opportunity created from the combination of factors described in the previous point, it is necessary to somehow measure the advantages and sacrifices one would sustain by pursuing that opportunity. To do that, one can use various methods, such as comparison tables relating to an opportunity's benefits, as well as analyzing analytic data that affects it directly or indirectly.

In this case, a SWOT analysis (a method normally used to assist the formulation of a strategy), presented in figure 4.3 was used to try and expose a more general vision of the advantages and disadvantages associated with taking the opportunity presented.

Figure 4.3: SWOT Analysis

CasperLabs elaborated a report on the state of the blockchain in 2023 (Labs 2023), based on a survey of 603 global business decision-makers spread evenly across the US, UK, and China. Some of the results of the survey were:

- Nearly 90% of businesses in the US, UK, and China are starting to use blockchain in some capacity.
- 87% are likely to invest in a blockchain solution in the next 12 months. This is especially true in China, where more than half are "very likely" to invest.
- 81% expect technology budgets to increase in 2023, even amid an expected downturn.

4.0.4 Idea Generation

Once the opportunity has been identified and analyzed, it is possible to start yielding ideas with the information obtained through the analysis process. With this activity is desired to generate a group of ideas, rather than one primary idea, to develop and maturate the opportunity.

The ideas to be modeled need to answer the following questions:

- Does the solution to be developed guarantee the immutability of the data records?
- Does the solution to be developed increase the security of attacks on the system as a whole?
- Does the solution to be developed facilitate the management of exclusivity contracts?

The projected ideas are the following:

1. Update the current system by adding a ledger mechanism that will track all the changes and transactions made by the users of the system;
2. The development of a Proof of Concept (PoC) for an MLS-like platform where it will be possible to analyze the management of exclusivity contracts by using smart contracts associated with a private blockchain system;

3. The development of a Proof of Concept (PoC) Blockchain network for demonstrating the usage of smart contracts in the management of exclusivity contracts.

4.0.5 Idea Selection

As explained in the previous chapters, blockchain proves to be an idea worth pursuing as it is presented as a strong solution to the problems listed before. The ledger mechanism could be a solution in terms of evidence for data tampering, however, it would be a centralized solution and the potential for automation that smart contracts enable would be missed. The second idea (development of a MLS platform), could potentially be the most comprehensive solution. However, it presents challenges in terms of its complexity, both in terms of the intricate business-related concepts it entails and the substantial time and knowledge required to create a reliable prototype.

The development of a PoC Blockchain network for demonstrating the usage of smart contracts in the management of exclusivity contracts is the optimal choice. It aligns with emerging technology trends, enhances transparency and security, leverages automation, offers scalability and flexibility, serves as a valuable reference point for future implementations, and provides a focused research contribution. All this while avoiding the complexity associated with a full MLS platform.

Following this idea, it is fundamental to decide which blockchain platform will be used to develop the PoC. From this, a Multi-Criteria Decision Making (MCDM) method can be used to, through mathematical equations, justify the chosen platform. The Analytical Hierarchy Process (AHP) is an MCDM method that decomposes a complex decision into a hierarchy descending from an overall objective to several levels from criteria, sub-criteria to the lowest level (Sudin et al. 2017). In this project, AHP will be used to obtain the weight of importance of the chosen criteria. This will later be used by the Technique of Order Preference by Similarity to the Ideal Solution (TOPSIS) method to determine the most appropriate blockchain platform given the project context.

4.0.6 Analytic Hierarchy Process (AHP)

The first phase of AHP is defining the criteria, figure 4.4 describes the hierarchy tree.

The chosen criteria are explained in depth:

- API Support - An enterprise system communicates with different systems and technical stacks. Furthermore, blockchain data can only be accessed through APIs due to its data storage mechanism. Therefore, API is one of the critical factors for developing an enterprise system.
- Security - Blockchain technology provides better security compared to traditional software applications. However, there are some significant differences between permissioned and permissionless blockchain networks. As is the case in this project, a permissioned blockchain is required.
- Performance - The performance of most of the blockchain platforms depends on their consensus algorithm. The permissioned nature of the blockchain to be used grants that performance will not be a major issue.
- Cost - An enterprise system has two types of cost, namely, initial cost and operational cost. The initial cost includes design and development costs. Operational cost includes

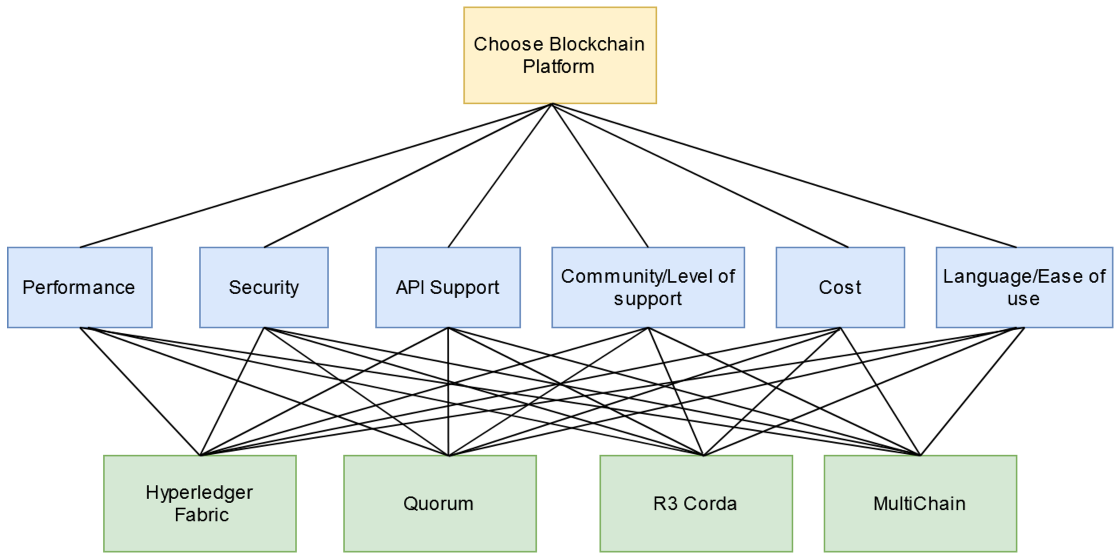

Figure 4.4: AHP Hierarchy Tree

software license fee, system maintenance fee, and platform usage charges. Gas price refers to the fee that the user has to pay for the usage of the blockchain platform. All this is bundled in this metric.

- Community/Level of Support - The level of support from a company or community is essential for the adaptation of new technology and its maintenance. This criterion includes direct support from the community that supports the technology as well as its market reputation.
- Language/Ease of Use - Language refers to the programming languages available to use with that platform. Ease of use was considered under two factors: present applications developed based on the platform and technical capabilities of the product.

The next step after the development of the structural hierarchy is to determine the priorities of elements at each level. The pairwise comparisons are based on how much more important element X is than element Y. The preference element is quantified using Saaty's ratio scale that is shown in 4.1.

Intensity of importance	**Definition**
1	Equal importance
3	Moderate importance of one over another
5	Essential importance or strong
7	Very importance strong
9	Extreme importance
2,4,6,8	Intermediate values between two adjacent judgments

Table 4.1: Saaty's Scale

To create the matrix, these names will be taken into consideration:

- **A** - API Support
- **S** - Security
- **P** - Performance
- **C** - Cost
- **CL** - Community/Level of support
- **L** - Language/Ease of Use

	P	**S**	**A**	**C**	**CL**	**L**
P	1	1/3	1/3	1/2	1/3	1/3
S	3	1	2	3	1	3
A	3	1/2	1	1	2	2
C	2	1/3	1	1	1	1/2
CL	3	1	1/2	1	1	1/2
L	3	1/3	1/2	2	2	1

Table 4.2: Comparison Matrix

This matrix shows that the criteria **S** are the most important due to the sum of its row being the highest value, on the other hand, **P** is the least important criterion.

The third AHP phase consists of the calculation of the relative priority (weight) of each individual criterion. To achieve this its mandatory to normalize the previous matrix and calculate the average of the values of each line of the normalized matrix. The normalization process is the result of the division of each value by the sum of its column.

	P	**S**	**A**	**C**	**Cl**	**L**	**Relative Priority**
P	0.0667	0.0952	0.0625	0.0588	0.0455	0.0455	0.0624
S	0.2000	0.2857	0.3750	0.3529	0.1364	0.0455	0.2932
A	0.2000	0.1429	0.1875	0.1176	0.2727	0.2727	0.1989
C	0.1333	0.0952	0.1875	0.1176	0.1364	0.0682	0.1230
CL	0.2000	0.2857	0.0938	0.1176	0.1364	0.0682	0.1503
L	0.2000	0.0952	0.0938	0.2353	0.2727	0.1364	0.1722

Table 4.3: Normalized matrix and relative priority

The fourth stage involves the calculation of the Consistency Ratio (CR) to ensure that a good level of consistency of the relative priorities is achieved. A value under 0.1 means that the relative priorities are consistent. The CR results in the division of the Consistency Index (CI) value by the Random Consistency Index (RI).

$$CR = \frac{IC}{IR} \tag{4.1}$$

- The Random Consistency Index (RI) is selected from the values presented in the table 4.4. Since the matrix dimension is 6, the IR value is 1.25.

1	2	3	4	5	6
0	0	0.52	0.89	1.11	1.25

Table 4.4: IR value table

- The IC value is calculated using the following formula:

$$IC = \frac{\lambda_{max} - n}{n - 1} \tag{4.2}$$

By multiplying the matrix in table 4.2 with the priority vector in 4.3, we get the following matrix:

P	0.3954
S	1.9142
A	1.3006
C	0.7808
CL	0.9391
L	1.1031

Table 4.5: Priority vector matrix

λ_{max} is returned by the getting the average of the matrix 4.3 divided by the priority vector.

$$\lambda_{max} = 6.40146857 \tag{4.3}$$

With the λ_{max} calculated, the CI value can be calculated, being n = 6:

$$IC = \frac{6.40146857 - 6}{6 - 1} IC = 0.06475 \tag{4.4}$$

Dividing the CI value by the RI value respective to size 6, the CR value is **0.06475**, which means that the relative priorities show good consistency (<0.1).

From these results, the weights of the criteria to be used in TOPSIS are the following:

- Security (**S**) - 0.2932
- API Support (**A**) - 0.1989
- Language/Ease of Use (**L**) - 0.1722
- Community/Level of Support (**CL**) - 0.1503
- Cost (**C**) - 0.1230
- Performance (**P**) - 0.0624

4.0.7 Technique of Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is a multi-criteria decision analysis method used to determine the best option from a set of alternatives based on a set of evaluation criteria. It was first introduced by Hwang and Yoon in 1981 (Hwang and Yoon 1981).

This method tests two alternatives: the ideal and the ideal negative. The ideal alternative is the one that presents a better level to the considered attributes/criteria. As for the ideal negative alternative, this shows the worst values for the considered attributes/criteria. TOPSIS selects the alternative that is closer to the ideal alternative and further from the ideal negative alternative. For this project, four blockchain platforms were considered to evaluate the one that better corresponds to the criteria: Hyperledger Fabric, Quorum, MultiChain, and R3 Corda. The criteria considered were the same as mentioned in the AHP section as the weights calculated previously were used in this method.

TOPSIS is more suitable for situations where the criteria are quantitative and the alternatives can be evaluated on a common scale. The 4 alternatives and 6 criteria previously established are presented in table 4.6.

Weights	0.0624	0.2932	0.1989	0.1230	0.1503	0.1722
	P	**S**	**A**	**C**	**CL**	**L**
Hyperledger Fabric	8	7	8	9	9	8
Quorum	6	7	7	3	9	6
MultiChain	5	7	8	2	5	7
R3 Corda	9	8	3	4	7	9

Table 4.6: Decision Matrix

The first step consists in normalizing the decision matrix. This is calculated following the formula:

$$r_{ij} = \frac{x_{ij}}{(\sum_i x_{ij}^2)^{\frac{1}{2}}} \quad \text{for} \quad i = 1, ..., m; j = 1, ..., n \tag{4.5}$$

Table 4.7 presents the normalized matrix.

	P	**S**	**A**	**C**	**CL**	**L**
Hyperledger Fabric	64	49	64	4	81	64
Quorum	36	49	49	64	81	36
MultiChain	25	49	64	81	25	49
R3 Corda	81	64	27	49	49	81
Sum	206	211	204	198	236	230
SquareRootSum	14.3527	14.5258	13.6382	14.0712	15.3623	15.1658

Table 4.7: Normalized Matrix

After obtaining the results in table 4.7, for each value x_{ij} in table 4.6, equation (4.5) is applied to obtain r_{ij}, which represents the normalized scores.

	P	S	A	C	CL	L
Hyperledger Fabric	0.5574	0.4819	0.5866	0.8581	0.5859	0.5275
Quorum	0.4180	0.4819	0.5133	0.2860	0.5859	0.3956
MultiChain	0.3484	0.4819	0.5866	0.1907	0.3255	0.4616
R3 Corda	0.6271	0.5507	0.2200	0.3814	0.4557	0.5934

Table 4.8: Matrix with r_{ij} values

The next step requires the calculation of the weighted normalized decision matrix, by multiplying the weights with each matrix value, with the formula:

$$v_{ij} = r_{ij} w_{ij} \tag{4.6}$$

It is time to select the ideal solution for each criterion in table 4.8, i.e. the highest value in the performance, security, API Support, Community/Level of support, Language/Ease of Use, and the lowest value in Cost. The following formula represents the ideal solution:

$$A^* = V_1^*, \ldots, V_n^*, \quad \text{where} \quad V_j^* = \max_i(V_{ij}) \quad \text{if} \quad j \in J; \min_i(V_{ij}) \quad \text{if} \quad j \in J' \tag{4.7}$$

Let J be the set of benefit attributes where more is better and J' the set of negative attributes where less is better. The ideal negative solution is represented by:

$$A^* = V_1^*, \ldots, V_n^*, \quad \text{where} \quad V_j' = \min_i(V_{ij}) \quad \text{if} \quad j \in J; \max_i(V_{ij}) \quad \text{if} \quad j \in J' \tag{4.8}$$

In table 4.9, the ideal solution is identified as green, and the ideal negative solution is identified as red.

	P	S	A	C	CL	L
Hyperledger Fabric	0.0348	0.1413	0.1167	0.0175	0.0880	0.0909
Quorum	0.0261	0.1413	0.1021	0.0700	0.0880	0.0681
MultiChain	0.0217	0.1413	0.1167	0.0787	0.0489	0.0795
R3 Corda	0.0391	0.1615	0.0438	0.0612	0.0685	0.1022

Table 4.9: Ideal solution and ideal negative solution

In this phase, it's necessary to calculate the separation between the ideal solution and the ideal negative solution.

This separation is the result of the following formula:

$$S_i^* = \left[\sum_j (V_j^* - V_{ij})^2 \right]^{\frac{1}{2}}, \quad i = 1, \ldots, m \tag{4.9}$$

Similarly, in order to reach the separation value of the ideal negative solution:

$$S_i' = \left[\sum_j (V_j' - V_{ij})^2 \right]^{\frac{1}{2}}, \quad i = 1, \ldots, m \tag{4.10}$$

In table 4.10, the values for the ideal positive and ideal negative are presented.

S^*	S'
0.0236	0.1062
0.0686	0.0709
0.0806	0.0738
0.0872	0.0506

Table 4.10: Separation from the ideal and ideal negative solution

The last step in TOPSIS consists of determining the relative proximity to the ideal solution, this is achieved by the formula:

$$C_i^* = \frac{S_i'}{S_i^* + S_i'}, \quad 0 < C_i^* < 1 \tag{4.11}$$

By applying this formula, we get the results displayed in table 4.11:

Hyperledger Fabric	0.8184
Quorum	0.5084
MultiChain	0.4779
R3 Corda	0.3670

Table 4.11: Calculation of the proximity to the ideal solution

To sum up this TOPSIS analysis, the three most important criteria were "Security", "API Support" and "Language/Ease of Use". The respective weight of each criterion results from the application of the AHP method. The TOPSIS method selects the alternative closer to the ideal solution and further from the ideal negative solution. By analyzing the table 4.11 it's easy to extract that Hyperledger Fabric presents a value closer to 1, which means that this is the closest to the ideal solution and further the ideal negative solution. From this analysis, we can conclude that the platform "Hyperledger Fabric" is the most appropriate platform for this problem.

4.0.8 Concept and Technology Development

Develop a PoC that through Blockchain technology can:

- Prevent data from being altered without the knowledge of the parties (tamper-evident)
- Automatize exclusivity contracts management with smart contracts

4.1 Value

The term "value" possesses a broad semantic variety, its meaning depends on the context in which is applied: economic, use, or perceived value. According to Miles "A product or service is generally considered to have good value if it has appropriate performance or cost. Or, by reverse definition, a product is considered not to have good value if it lacks either appropriate performance or cost.". Miles also states (L.D. Miles 2015):

1. Value is always increased by decreasing costs (while, of course, maintaining performance).
2. Value is increased when performance is also increased. If the customer needs, wants, and is willing to pay for more performance.

Other authors, such as Schechter, have suggested that viewing value as a trade-off between only quality and price is too simplistic (Schechter 1984). Porter talked about providing “superior value to the buyer in terms of product quality, special features, or after-sale service.” (Porter 1990). These views suggest that existing value constructs are too narrow and that dimensions other than price and quality would increase the construct’s usefulness.

In terms of consumer value, Morris B. Holbrook defines it as an "interactive relativistic preference experience" (Holbrook 1996). Holbrook also presents the principles of this concept (Holbrook 1996):

- Interactive - It requires an interaction between some subject (a customer) and some object (a product). For example, a person driving a car is an interaction between the user and the product.
- Relativistic - Value is situational meaning that it is not universally held or objective, but rather varies depending on the individual and their context.
- Preference - When choosing a product, the customer transmits his preconceived values to participate in the evaluative judgment. If something is good/bad, if the customer likes/dislikes, that will lead the customer to build its opinions about the pros and cons.
- Experience - The culmination of the previous points; customer value resides not in the purchase itself but rather in the consumption experience. This claim is inherent in the concept of an "interactive relativistic preference".

To measure the product, in terms of value produced, Holbrook identifies the following metrics (Holbrook 1996):

- Efficiency - Efficiency can be an important part of a product’s functional value, especially in situations where time, money, or resources are limited. For example, a washing machine that can wash clothes quickly and with minimal water and electricity usage provides functional value in terms of efficiency.
- Excellence - In this type of value, the customer admires some object for its capacity to serve as the means to complete a task, fulfill a purpose, or user experience as a value.
- Status - As stated by the term "status", it designates when a customer shifts its consumption behavior toward the other-oriented end of achieving a favorable response from someone else.
- Esteem - Intrinsic relation to status, but this time the consumer doesn’t desire a favorable response from someone else, as the act of owing is itself a source of satisfaction.
- Play - This attends to the fun provided by the use of the product/service.
- Aesthetics - Self-oriented appreciation of some object where this experience is valued as an end in itself. For example, an aesthetically pleasing design of a car can add to the overall enjoyment of driving it.

- Ethics - This relates to the consequences of the use of a product in terms of what it will cause to others and their reaction to it.
- Spirituality - Equivalent to ethics and is associated with the belief from customers that products or services that promote a sense of inner peace, well-being, or association with the divine return emotional benefits.

In the context of this project, it will take into account the following: efficiency, excellence, esteem, and aesthetics. Efficiency because with fewer steps the client can get data backup and contract management process automatically due to inherent mechanisms of the Blockchain such as the ledger and smart contracts. Excellence, as the goal of this project, is to satisfy an already existing need. Esteem, since the MLS prototype is supposed to solve a trust problem for the agents involved in the system. And finally, Aesthetics, where in addition to its design, the user will use the service for its inherent value.

Zeithaml has suggested that perceived value can be regarded as a "consumer's overall assessment of the utility of a product (or service) based on perceptions of what is received and what is given."(Zeithaml 1988). If the benefits have more weight than the sacrifices, the solution has value to the customer. Different customer segments can have different value perceptions about the same product or service.

To evaluate the value for the customer was created Table 4.12 was with the benefits and sacrifices associated with this project.

Benefits	Sacrifices
- Improved data security - Automation of contract management	- Adaptation to a new system - System's cost - Need to make changes to existent processes

Table 4.12: Benefits and Sacrifices to the customer

The system is capable of securing data more efficiently due to the inherent blockchain ledger transparency and immutability properties. Due to smart contracts technologies, the customer gains a new way to manage contracts, automation can be an advantage in terms of efficiency and excellence. The only drawbacks are the cost related to the initial investment and normal system usage as well as possible adjustments in existing processes.

4.1.1 Value Proposition

The Value Proposition Canvas makes explicit how value is created for the customers. It helps the company design products and services that their customers want (Osterwalder et al. 2015).

Proposed by Osterwalder, the Value Proposition Canvas consists of two main components: the customer profile and the value map.

The customer profile describes the jobs-to-be-done, pains, and gains:

1. Job to be Done are the objectives that the customers are trying to meet with their current operations, and that the product/service presented in the Value Proposition is going to solve.

2. Gains are the benefits that the customer desires and that are nice for them to see in the product/service, sometimes even as extras;

3. Pains are the difficulties the customer feels and goes through in the process of achieving the result aimed by the product/service;

The value map outlines the products and services being offered, and how they create value for the customers:

1. Product & Services - What products/services is the value proposition based on?

2. Gain Creators - How is the product/service delivering gains?

3. Pain Relievers - How the product/service will "relieve" customer pains (problems).

The value proposition for this project is presented in figure 4.5

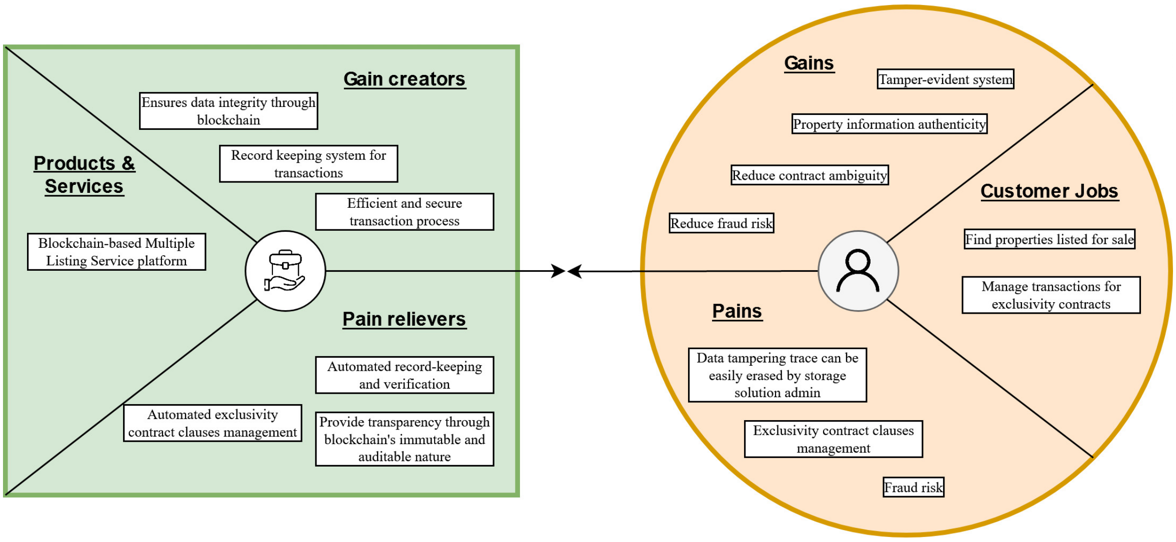

Figure 4.5: Value Proposition Canvas

Considering the intent of the solution being described in the context of this project, the customer segment addressed consists of real estate agents/brokers that utilize a traditional centralized Multiple Listing Service.

4.1.2 Quality Function Deployment

Quality Function Deployment (QFD) is a system for designing a product or service based on customer demands as well as the functionalities to identify the requirements, House Of Quality synthesizes the relation between these (Larry Miles, Dahlgaard-Park, and Group 2007). The purpose of QFD is divided into three:

1. Launch products or services to market faster and at a lower cost

2. Achieve customer-driven product design

3. Provide a tracking system for future process improvements

Using this technique is expected to get a better understanding of consumer needs, improve the organization on development projects, mitigate production problems, prepare the product to have fewer conceptual changes, get a reputation for quality assurance, get an overall better success rate, and to provide the whole team a good documentation about the ongoing work.

The House of Quality (HOQ), also known as the product planning matrix, is an important stage of the QFD:

The presented HOQ in figure 4.6 is characterized by the Demanded Quality, also known as Consumer Requirements and "Whats", and by the Quality Characteristics, also known as Functional Requirements and "Hows". The list below enumerates all the demanded qualities, as well as their importance:

- Data Security (8)
- Automate payment processing of exclusivity contracts (7)
- Security (6)
- Friendly UX (5)
- Error Prevention (5)
- System Availability (6)
- Identity Management (7)
- Property Management (8)

To match the consumer requirements there are the functional requirements, that explain how the expected additional value will be created:

- Blockchain Implementation
- Private access to the blockchain
- Database Implementation
- Smart contracts implementation
- Intuitive user interface
- Authentication and Authorization Service

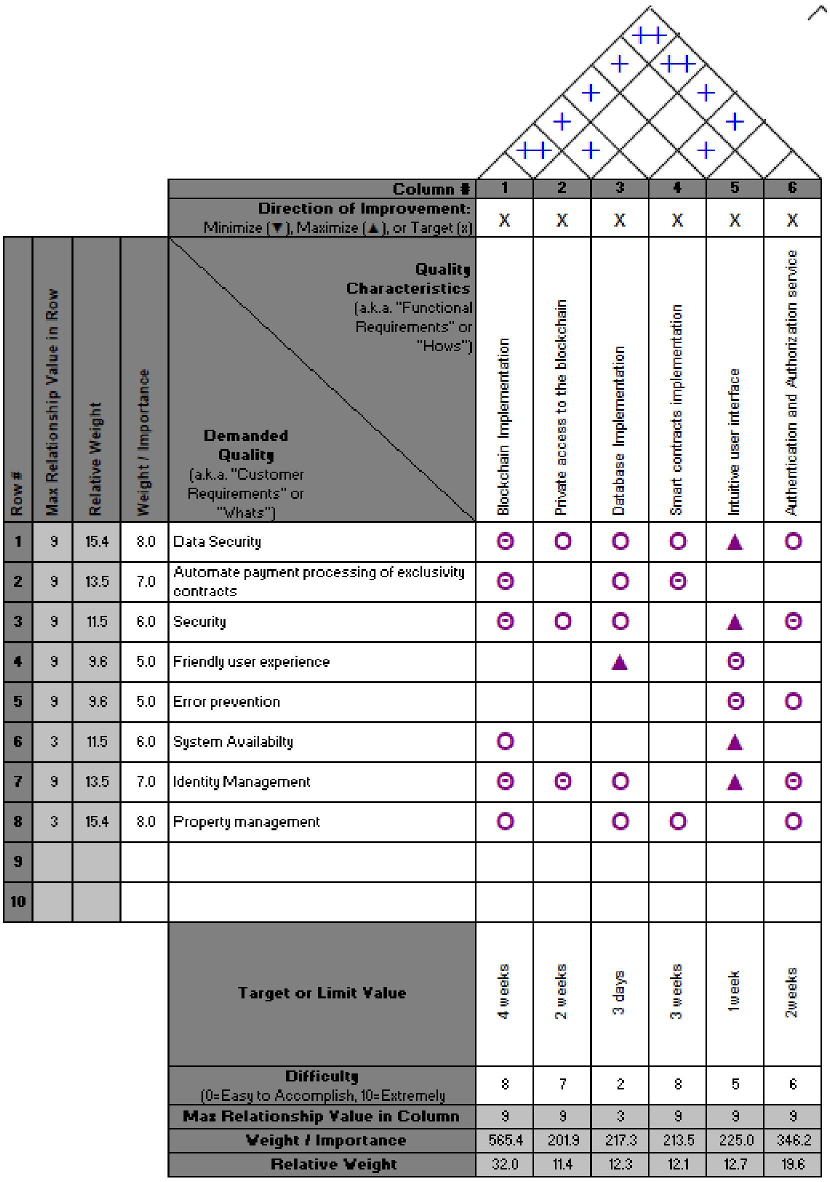

Figure 4.6: House of Quality

Chapter 5

Solution

This chapter explores the design of the proof-of-concept application.

Building upon the analysis conducted in the previous chapter, it proceeds to delineate the software development process across three distinct dimensions: firstly, the specification of software requirements; secondly, the domain-level design; and lastly, the architectural design.

5.1 Software Requirements Specification

Software requirements encompass the specification of the anticipated actions and operations of the software to be developed. This is a process within software engineering that studies the formulation, analysis, development and maintenance of the requirements (including both functional and non-functional) that the system must fulfill to address a given issue.

This section includes:

1. The identification of the actors participating in the system.
2. The description of the functional requirements that present the interaction between these actors and the system
3. The non-functional requirements that represent the constraints on the design or implementation, such as data security and auditability, among others.

5.1.1 Actors

For this proof-of-concept application, there is one main actor.

Real estate agents are the direct actors. Agents submit/update/delete contracts.

5.1.2 Functional Requirements

Functional Requirements are portrayed as the functionalities that the software needs to provide. These encompass the features that enable the system to operate as it is intended. In Figure 5.1, a use case diagram illustrates the functional requirements extracted from the software requirements specification for the Proof of Concept (PoC).

The functional requirements are detailed below:

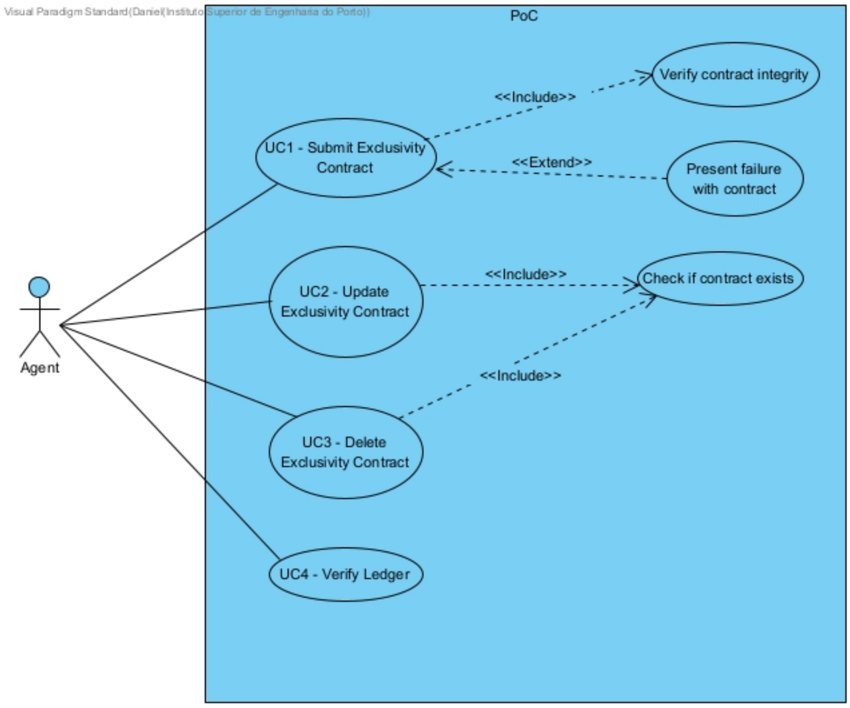

Figure 5.1: Use Case Diagram

Requirement nº	**1**
Name	Submit Exclusivity Contract
Description	Agent submits the exclusivity contract to be approved and submitted
Actor	Agent
Precondition	The contract must contain all the necessary details and it can't exist another exclusivity contract for the same property
Postcondition	The contract is registered and the new transaction details appear in the ledger

Table 5.1: Requirement 1 - Submit contract

Requirement nº	**2**
Name	Modify Exclusivity Contract
Description	The Agent emits an order to modify the conditions of a previously submitted (and approved) contract
Actor	Agent
Precondition	A contract related to this property must have been submitted previously
Postcondition	The contract is registered and the new transaction details appear in the ledger

Table 5.2: Requirement 2 - Modify Exclusivity Contract

Requirement nº	3
Name	Cancel Exclusivity Contract
Description	The Agent emits an order to rescind an exclusivity contract
Actor	Agency
Precondition	A contract related to this property must have been submitted previously
Postcondition	The contract is revoked and the new transaction details appear in the ledger

Table 5.3: Requirement 3 - Cancel Exclusivity Contract

Requirement nº	6
Name	Verify Ledger
Description	All the parties involved can verify the ledger to track the records of exclusivity. Including the ones who are being executed and those that were canceled
Actor	Agency
Precondition	The agent must have authorization to verify the ledger
Postcondition	-

Table 5.4: Requirement 6 - Verify Ledger

5.1.3 Non-functional requirements

Non-functional requirements play a vital role in the software development process, as they delineate the system's quality attributes.

In the context of this project, certain considerations arise regarding the confidentiality and safeguarding of data, which are presented in table 5.5.

Requirement	Description
Access Control	Only the parties involved in the network have permission to access the data.
Auditability	The application must have the capacity to show what happened, who did it, and when it was done.
Scalability	Ensure that the system is prepared to increase its availability as nodes increase

Table 5.5: Non-Functional Requirements

General Data Protection Regulation (GDPR) aspects are not taken into account within this PoC application, as addressing these concerns would require collaboration with a legal expert. Consequently, the integration of GDPR measures is designated as future work.

5.2 Domain Modeling

The domain model is a conceptual representation that portrays the entities within a domain, their respective attributes, and the connections between them. It is used to communicate with non-technical stakeholders as it employs the domain vocabulary. In Unified Modelling

Language (UML), a class diagram represents the domain model. It serves as a means to engage non-technical stakeholders by employing domain-specific terminology. In the context of Unified Modeling Language (UML), the domain model is typically visualized using a class diagram.

Figure 5.2 represents the domain model of the PoC application. It presents the three entities and their corresponding properties within the domain. These entities include:

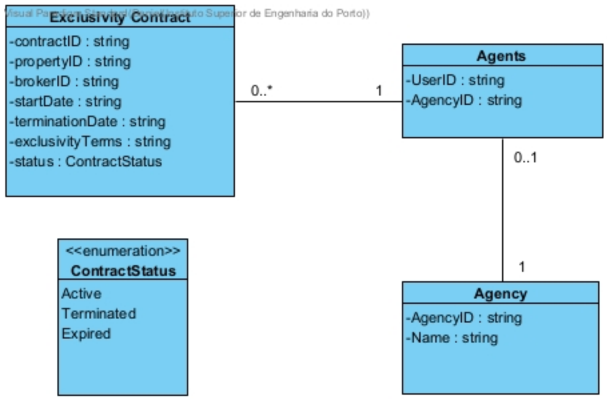

Figure 5.2: Domain Model

- Exclusivity Contract is the central entity of this domain and represents the assets that are tracked in the Hyperledger Fabric channel.
  - Contract ID: A unique ID that identifies an exclusivity contract.
  - Property ID: A unique ID that identifies a property, it could be seen as the ID used in the universe of registered properties such as the Portuguese equivalent "Número de Propriedade Predial" (NPI).
  - Broker ID: A unique ID that identifies a broker or agent.
  - Start Date: The date on which the exclusivity contract becomes legally binding and enforceable. From this date, the terms and conditions outlined in the contract are applicable.
  - Termination Date: The date on which the exclusivity contract concludes, and the rights and obligations of the parties defined in the contract cease to be in effect. After this date, the contract is considered expired.
  - Status: The status of the contract. It can be considered:
    * "Active" - It is currently in effect and all the agreed-upon terms and conditions are being followed.
    * "Expired" - When the agreed-upon period for the contract has ended, and the parties are no longer bound by its terms.
    * "Terminated" - When it is ended prematurely before the agreed-upon terms are fulfilled, often due to a violation, breach, or mutual agreement of the involved parties.
- Agents
  - User ID: A unique ID of the User registered in the network.

  - Agency ID: A unique ID of the Agency that the agency is associated to
- Agency
  - Agency ID: The unique ID of the agency registered in the network.
  - Name: The name of the agency.

5.3 Hyperledger Fabric Network Design

A network implemented with Hyperledger Fabric usually contains the following components:

- Peer - A node in the network that maintains a copy of the ledger and conducts transactions via chaincode. There are three different types of peers.
  - Endorsing Peers - These peers simulate and endorse transactions by executing chaincode. Their endorsements validate transactions before they are sent to the ordering service.
  - Committing Peers: After transactions are ordered and endorsed, committing peers validate and commit these transactions to the ledger, ensuring its consistency across the network.
  - Anchor Peers: Anchor peers are used for cross-organization communication. They maintain connection information for peers in their own organization and other organizations.
- Channel - A private subnet within a Hyperledger Fabric network that allows a group of participants to execute transactions and share data in a confidential manner. Each channel has its own separate ledger.
- Chaincode - Encapsulates the business logic of the blockchain network. It specifies the rules for updating the ledger and determines which transactions are valid. When a transaction is submitted to the network, it is sent to the appropriate peer nodes for endorsement. The chaincode is then executed on the endorsing peer nodes and the endorsed transaction is sent back to the client for ordering and finally commit to the ledger.
- Ledger - A distributed database that records all of the transactions that occur on the network. Each peer node maintains a copy of the ledger, and the ledger is updated whenever a new transaction is endorsed and committed to the network. The ledger is composed of two parts: the world state, which stores the current state of all assets on the network, and the transaction log, which stores a record of all transactions that have occurred on the network. Hyperledger offers the option to use CouchDB as a state database, which is going to be implemented in this project.
- Orderer - Responsible for ensuring the delivery of transactions to the appropriate peer nodes for validation and endorsement. The orderer maintains an ordered log of all transactions that have occurred on the network and provides a communication channel for the peer nodes to reach a consensus on the order in which transactions should be processed. There are 5 types of Orderer, however, in this PoC the Raft orderer type was chosen as it is the most reliable and common to be used.
- Membership Service Provider (MSP) - Defines the rules for identity management and authentication within a Hyperledger Fabric network. MSPs are used to verify the

identity of participants in the network, such as users, applications, and peer nodes. This component ensures the security and integrity of the network by making sure that only authorized entities are able to access the network and participate in transactions. For this PoC the type selected was Certificate Authority (CA) based.

- Clients - Initiate transactions by sending proposals to endorsing peers. They interact with the network using software development kits (SDKs) and REST APIs.

The high-level request flow of a transaction is presented in Figure 5.3. The lifecycle of a transaction encompasses the following steps:

1. The client connects to a Hyperledger Fabric network, creates a transaction and sends it to the endorsing peer.
2. The endorsing peer verifies the client's signature, simulates a transaction and if every condition set initially checks, the peer sends an endorsement signature.
3. If the transaction is endorsed, the client (indirectly) submits the transaction to the ordering service. Otherwise, the transaction is canceled.
4. The ordering service delivers a transaction to the peers. All peers commit and apply the same sequence of transactions and update their state.

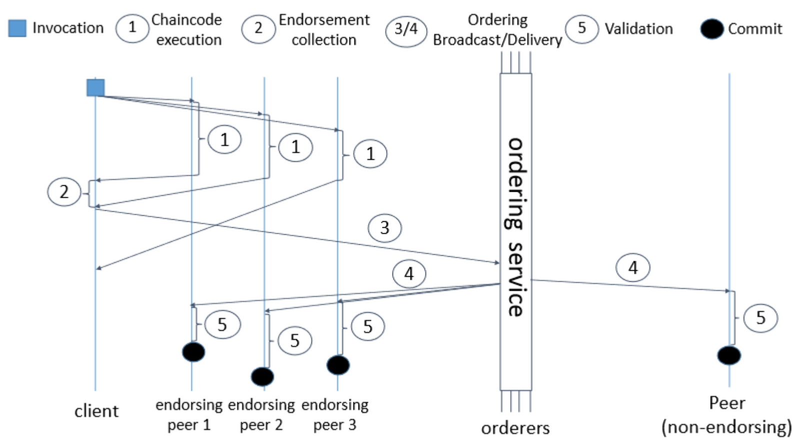

Figure 5.3: Hyperledger Fabric high level transaction flow (Androulaki et al. 2018)

5.4 Architectural Design

Having established a comprehensive understanding of the fundamentals of Hyperledger Fabric architecture, the focus now shifts towards proposing a solution architecture tailored to the management of real estate exclusivity contracts.

Leveraging the key insights gained from the previous sections, the proposed solution architecture is designed to address the requirements defined for this Proof-of-Concept application.

5.4.1 Solution Architecture Overview

Figure 5.4 proposes the network's architecture to be implemented for this PoC. The components presented in the diagram are:

- Two Organizations, Agency 1 and Agency 2, representing a real estate agency. Both share the same channel (mls-channel) and, consequently, the same ledger.
  - Each of the participating organizations runs a peer node that maintains information about its local copy of the distributed ledger in a dedicated CouchDB database node and hosts the chaincode to be executed.
  - Each organization contains an MSP folder containing the certificates that enable performing actions on the network.
- An Orderer node, running the Raft consensus algorithm, ensures that transactions are delivered to the correct channels and to committing peers.
- The Client Application node is a registered user that acts as a client in the network, making requests to the application's API that will, via its agency Peer, execute chaincode interactions with the channel's ledger.

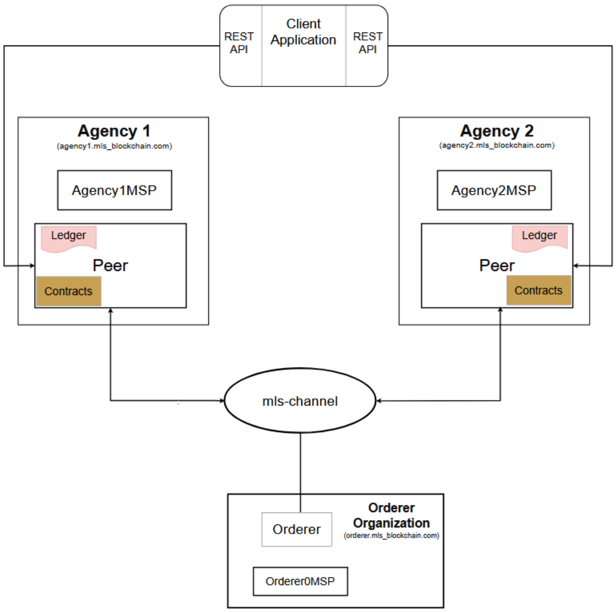

Figure 5.4: PoC Network Architecture

5.4.2 Deployment Process

This section presents the process of deploying the network, whose architecture was shown in the previous section. For this environment a local Kubernetes (Rensin 2015) cluster is used in order to emulate the distributed nature of the whole system. An NFS server is used to store files such as certificates, identities, chaincode and channel configurations.

Some of the reasons to use this deployment strategy using Kubernetes include (Yewale 2018):

- Hyperledger Fabric is built into container images. Kubernetes provides us with tools to automate deployment, scaling and other management of containerized applications.
- Hyperledger Fabric components can achieve high availability due to a Kubernetes feature that monitors running pods and brings up crashed ones automatically.

- Kubernetes supports multi-tenancy, meaning that multiple isolated Fabric instances can run on the same Kubernetes platform.

Even though this application serves as a Proof of Concept, deploying its components within a cloud environment through a local Kubernetes cluster showcases its readiness for migration to a scalable production environment.

Each component of the system runs on a Docker container inside a Kubernetes Pod that is managed by a Kubernetes Deployment. Each Pod is exposed to the remaining components of the cluster through a specific Kubernetes Service. The Docker containers use the official Hyperledger Fabric images for each component. Figure 5.5 presents a deployment diagram of the network.

Peer, CA and Orderer deployments will have access to the NFS shared storage as these need the artifacts stored in there.

Figure 5.5: Complete application deployment diagram

Minikube was used to create the local Kubernetes cluster, as it is designed to enable developers to develop and test applications on their local computers before deploying them to larger Kubernetes clusters (Zahoor 2023). Figure 5.6 presents the Minikube control dashboard of the local cluster. Besides the deployments mentioned before:

- Jobs are a resource used to create and manage a task that runs to completion, in this case, the tasks are the creation of the channel and its artifacts
- Replica Sets are a resource that ensures a specified number of replica Pods are running at any given time, it is used to guarantee the availability and scalability of the application.

In order to deploy the network on Kubernetes, configuration files first needed to defined. Following is the process involved in deploying the blockchain network on Kubernetes using .yaml files:

1. Create Persistent Volume (in this case connect it to a shared folder in the NFS Server) and Persistence Volume Claim.
2. Deploy the CA Servers for Agency1, Agency2 and Orderer.
3. Run the certificates (job.yaml) that will use the CA to generate the MSP folder for the orderer and peers.
4. Deploy a Job in order to run scripts to generate the channel artifacts. The configtx.yaml file is essential as it contains the configuration parameters for the Hyperledger Fabric network.
5. Deploy the Orderer, peers (peer and cli-peer).
6. Create services for all peers, ca, orderer.
7. Channel operations:
    - Create App channel (mls-channel), by running the peer command.
    - Join the peers (from both organizations) to the mls-channel.
    - Update Anchor Peers, which is not relevant in this case as each organization only has one peer.
8. Install chaincode on each peer.
    - Package the chaincode, this will generate a .tgz file that will be related to each organization.
    - Install the chaincode in each peer using the generated .tgz file, this returns a package ID that will identify the chaincode in the peer.
9. Build Docker image of the chaincode, create chaincode deployment and service.
10. Approve chaincode on each organization by using the peer command
11. Create API and Hyperledger Explorer deployment and service.

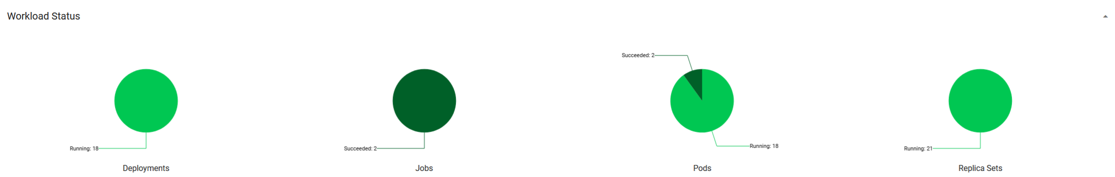

Figure 5.6: Minikube Dashboard

In order to interact with the network, a basic frontend was implemented using the Angular framework to be used as a back office application for the management of contracts. This frontend communicates with a Node.js API, which, in turn, connects to the network's peers. Through this architecture, the frontend initiates requests, enabling execution of query and modification transactions on the network.

The application flow, depicted in Figure 5.7, unfolds as follows: The client initiates interaction with the application web server to invoke the smart contract. Subsequently, a Certificate Signing Request (CSR) is generated to validate the client's identity eligibility for the request. Upon successful verification, the API establishes a connection with the network peer and executes the client's request.

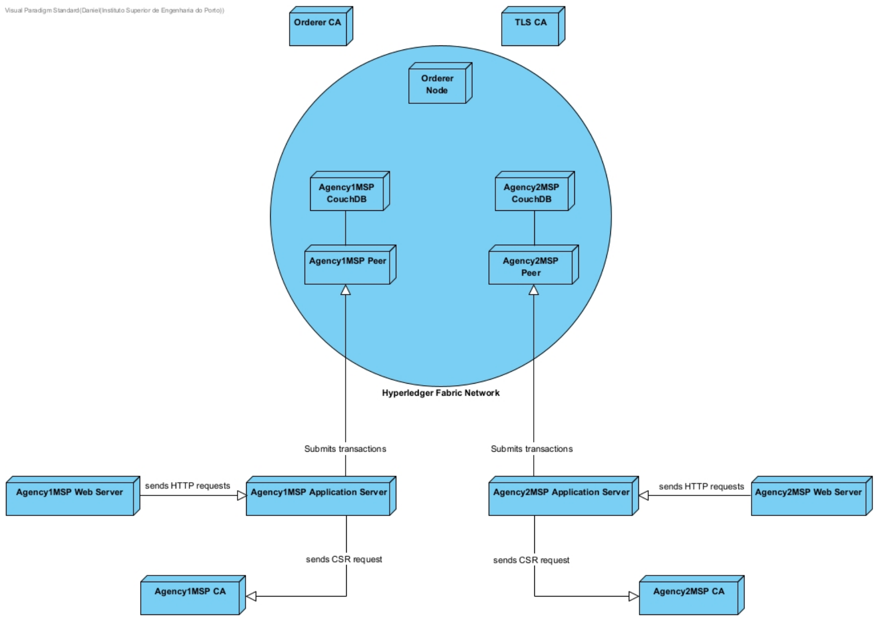

Figure 5.7: System components architecture

Figure 5.8 showcases the primary dashboard of Hyperledger Explorer, an official tool provided by Hyperledger Fabric. This tool enables the visualization of various network components, including transactions, blocks, channels, chaincodes and organizations that form an integral part of the network infrastructure.

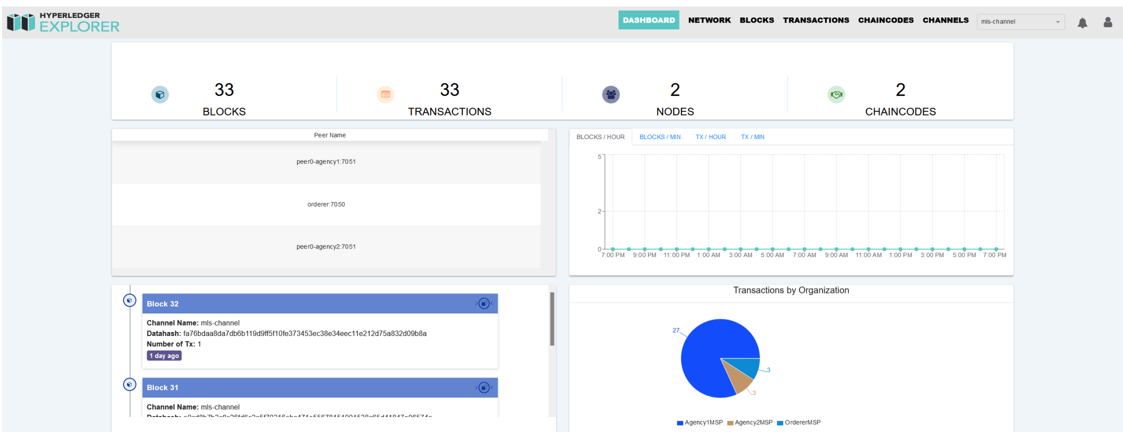

Figure 5.8: Hyperledger Explorer Dashboard

5.4.3 Implementation

This section presents the details of the contract management application. The corresponding code and configuration files can be found on GitHub:

https://github.com/danielfmdias/Blockchain-and-Property-Technology.git

In the development of this Proof of Concept (PoC) application, three essential components were addressed:

- Establish the Hyperledger Fabric network.
- Define the API to connect with the network peers.
- Develop a user-friendly frontend to interact with the network.

5.4.4 Hyperledger Fabric network implementation

Part of the configuration process of an Hyperledger Fabric network is the definition of network details via the configtx.yaml file. This file acts as the blueprint, detailing the network configuration parameters essential for creating a new consortium, channel, or organization within the Fabric network.

As illustrated in Figure 5.9 and 5.10, a snippet of the configtx.yaml file is shown below. Figure 5.9 shows an example of the "Organizations" section, where is defined the different organizational identities which will be referenced later in the configuration. In this section, we have the following parameters:

- Name: Specifies the name of the organization as Agency1MSP.
- ID: Represents the identifier for the MSP configuration, set as Agency1MSP.
- MSPDir: Indicates the filesystem path containing the MSP configuration for this organization.
- Policies: Defines the access control policies for this organization. For example, Readers policy allows entities with "Agency1MSP.member" role to read from this organization.
- AnchorPeers: Specifies the location of peers that can be used for cross-organizational gossip communication. In this case, there is one anchor peer specified with the hostname peer0-agency1 and port 7051.

```yaml
    - &Agency1
        # DefaultOrg defines the organization which is used in the sampleconfig
        # of the fabric.git development environment
        Name: Agency1MSP

        # ID to load the MSP definition as
        ID: Agency1MSP

        MSPDir: ../organizations/peerOrganizations/agency1.mls_blockchain.com/msp

        # Policies defines the set of policies at this level of the config tree
        # For organization policies, their canonical path is usually
        #   /Channel/<Application|Orderer>/<OrgName>/<PolicyName>
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Agency1MSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('Agency1MSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('Agency1MSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('Agency1MSP.peer')"

        # leave this flag set to true.
        AnchorPeers:
            # AnchorPeers defines the location of peers which can be used
            # for cross org gossip communication.  Note, this value is only
            # encoded in the genesis block in the Application section context
            - Host: peer0-agency1
              Port: 7051
```

Figure 5.9: Agency1 organization definition in configtx.yaml

Figure 5.10 shows an example of the Profiles section, where different configurations for different network setups are defined. Profiles specify how the network components (orderers, peers, organizations, etc.) are configured and interact within the network. Going in more detail:

- TwoOrgsOrdererGenesis Profile:
  - Defines configuration for the initial block (genesis block) of the blockchain.
  - Specifies orderer type (consensus) as 'raft' and sets up orderer nodes.
  - Lists participating organizations (only OrdererOrg in this case).
- TwoOrgsChannel Profile:
  - Defines configuration for a specific channel (TwoOrgsChannel).
  - Specifies the channel's consortium (SampleConsortium).
  - Lists participating organizations (Agency1 and Agency2).

```yaml
Profiles:
    TwoOrgsOrdererGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OrdererDefaults
            OrdererType: etcdraft
            EtcdRaft:
                Consenters:
                  - Host: orderer.mls_blockchain.com
                    Port: 7050
                    ClientTLSCert: ../organizations/ordererOrganizations/mls_blockchain.com/orderers/orderer.mls_blockchain.com/tls/server.crt
                    ServerTLSCert: ../organizations/ordererOrganizations/mls_blockchain.com/orderers/orderer.mls_blockchain.com/tls/server.crt
            Organizations:
                - *OrdererOrg
            Capabilities:
                <<: *OrdererCapabilities
        Consortiums:
            SampleConsortium:
                Organizations:
                    - *Agency1
                    - *Agency2
    TwoOrgsChannel:
        Consortium: SampleConsortium
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *Agency1
                - *Agency2
            Capabilities:
                <<: *ApplicationCapabilities
```

Figure 5.10: Profiles section in configtx.yaml

The chaincode implementation is also part of establishing the network. This component represents the core of the system's business logic. The chaincode named "contracts" was implemented using the Node.js Fabric SDK. The chaincode was created to meet the following exclusivity contract eligibility principles: - One "propertyID" for one Contract. - The Contract dates should be up to date meaning that when a Contract has a "terminatedDate" parameter before the current date it's considered "expired".

Figure 5.11 shows a class diagram of the chaincode named "contracts".

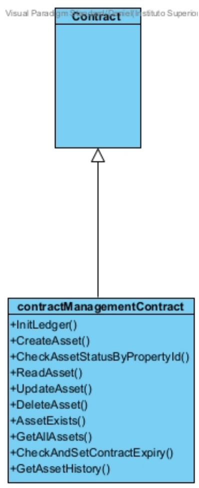

Figure 5.11: Chaincode class diagram

Going over the functions that exist in this smart contract:

- InitLedger: The InitLedger function initializes the ledger with a predefined set of contracts. It creates three sample contracts and stores them in the world state using their unique IDs. The contract data model was mentioned in the section 5.2.
- CreateAsset: The CreateAsset function creates a new asset (contract) in the world state. It checks if the asset with the given ID already exists in the world state. Also, invoking the CheckAssetStatusByPropertyId function verifies the possibility of having an existing contract with the same "propertyID":
  - If the contract status is "active", the contract is not created.
  - If the contract status is "expired" or "terminated", an error is raised saying that the contract already exists and recommends the user change the status of the old one.
- CheckAssetStatusByPropertyId: The CheckAssetStatusByPropertyId function checks if an asset with a specific "propertyID" exists and returns its status (Active, Expired, or Terminated).
- ReadAsset: The ReadAsset function retrieves the asset details based on the given ID from the world state.
- UpdateAsset: The UpdateAsset function updates an existing asset with new information. It first checks the asset's existence and then overwrites the existing data with the new data provided.
- DeleteAsset: The DeleteAsset function deletes an asset from the world state based on the given ID.
- AssetExists: The AssetExists function checks if an asset with a specific ID exists in the world state.
- GetAllAssets: The GetAllAssets function queries the world state to retrieve all assets in the ledger.

- CheckAndSetContractExpiry: The CheckAndSetContractExpiry function verifies if a contract is expired by checking its "terminationDate". In case that happens, change the contract status to "expired".
- GetAssetHistory: The GetAssetHistory function retrieves the historical transactions related to a specific asset.

5.4.5 API implementation

The Application Programming Interface acts as a crucial bridge between the clients and the underlying blockchain network. Its primary purpose is to enable communication between users and the blockchain peers. By providing a simplified and standardized interface, the API abstracts the complexities of blockchain interactions, making it user-friendly and accessible.

The API was designed using a RESTful architecture, leveraging HTTP methods for communication. It integrates with the blockchain peer using the Node.js SDKs (Software Development Kits) provided by Hyperledger Fabric (Hyperledger 2023b). This SDK provides the following libraries:

- fabric-ca-client: to interact with the fabric-ca to manage user certificates.
- fabric-common: encapsulates the common code used by all fabric-sdk-node packages supporting interactions with the Fabric network to send transaction invocations.

The API is built using Node.js and Express.js. It follows the MVC (Model-View-Controller) pattern for organized code structure. The endpoints included in the API are the following:

- Register User in the network (/register)
- Add a Contract in the network (/createAsset)
- Update a Contract (/updateAsset)
- Delete a Contract (/deleteAsset)
- Get All Contracts (/getAllAssets)
- Get the history of transactions associated with a Contract (/getAssetHistory)

Figure 5.12 illustrates the code for the register user endpoint. The JSON request body includes the user's affiliated organization and its userID. The API verifies if a MSP exists for the input organization and then proceeds to the registration of that user.

```
app.post("/register", async (req, res) => {

    try {
        let org = req.body.org;
        let userId = req.body.userId;
        let result = await registerUser({ OrgMSP: org, userId: userId });
        res.send(result);

    } catch (error) {
        res.status(500).send(error)
    }
});
```

Figure 5.12: Register user endpoint

Figure 5.13 and 5.14 illustrate the code for the creation of an asset (exclusivity contract). Relatively to the previously mentioned endpoint, it concatenates to the JSON request the channel used for the transaction, the chaincode to be invoked and the data containing the contract data. Then, the gateway requests the identity of the user and channel in order to call the method from the chaincode.

```
app.post("/createAsset", async (req, res) => {
    try {


        let payload = {
            "org": req.body.org,
            "channelName": channelName,
            "chaincodeName": chaincodeName,
            "userId": req.body.userId,
            "data": req.body.data
        }

        let result = await createAsset(payload);
        res.send(result)
    } catch (error) {
        res.status(500).send(error)
    }
})
```

Figure 5.13: Create asset endpoint

```
exports.createAsset = async (request) => {
    let org = request.org;
    let num = Number(org.match(/\d/g).join(""));
    const ccp = getCCP(num);

    const wallet = await buildWallet(Wallets, walletPath);

    const gateway = new Gateway();

    await gateway.connect(ccp, {
        wallet,
        identity: request.userId,
        discovery: { enabled: true, asLocalhost: true } // using asLocalhost as this gateway is using a fabric network deployed locally
    });

    // Build a network instance based on the channel where the smart contract is deployed
    const network = await gateway.getNetwork(request.channelName);

    // Get the contract from the network.
    const contract = network.getContract(request.chaincodeName);
    let data=request.data;
    //let result = await contract.submitTransaction('CreateAsset',data.ID,data.color,data.size,data.owner,data.appraisedValue);
    let result = await contract.submitTransaction('CreateAsset',data.ID,data.brokerId,data.propertyId,data.startDate,data.terminationDate,data.terms,data.status);
    return (result);
}
```

Figure 5.14: Create Asset chaincode call

5.4.6 Frontend Implementation

To enhance the user experience, a CRUD interface (5.16) was implemented using the Angular framework. The choice of Angular was deliberate, influenced by the author's familiarity with the technology. In addition to the basic CRUD operations, it allows for verification of the history of an asset. If an asset is updated, it is possible to see the different object states with the timestamps and transaction IDs in order to help track these changes when verifying the transactions in Hyperledger Explorer.

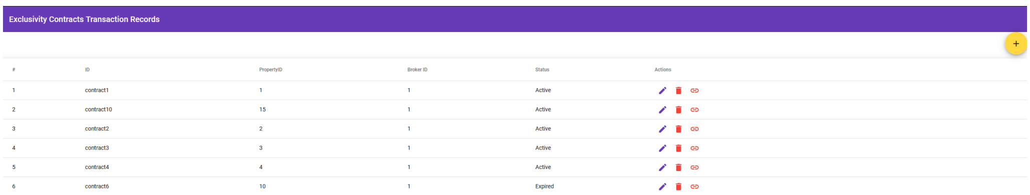

Figure 5.15: Contract list view

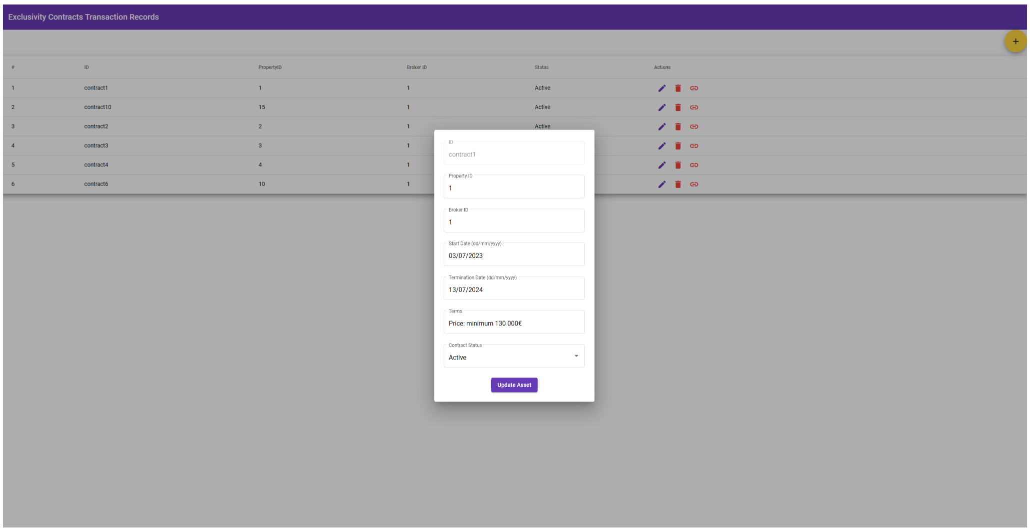

Figure 5.16: Update a contract form view

Integration with the Fabric network is established through a set of the API endpoints mentioned in the last subsection. The communication primarily happens via HTTP requests, typically using the RESTful architecture style, ensuring a standard and stateless interaction between the frontend and backend.

Figure 5.17 presents a code snippet of the HttpService class. For simplicity reasons, and due to the UI not being the focus of the PoC, the organization and userID are hardcoded. The environment.baseURL variable points to the IP of the API service deployed in Kubernetes (http://127.0.0.1:4000/)

```
updateAsset(result: any) {
  let data = {
    org: "Agency1MSP",
    userId: "User2",
    data: result
  }

  console.log(data)
  return this.http.post(`${environment.baseUrl}updateAsset`, data)
  .pipe(
    catchError(this.handleError)
  );
}
```

Figure 5.17: Update method HTTP call

Chapter 6

Tests & Solution Evaluation

Part of the software development process involves assessing whether the software meets its intended objectives and goals within a specific timeframe.

Ensuring the correct functioning of all blockchain components while identifying and rectifying any defects in the software is vital. This process establishes trust among users. The upcoming section outlines tests designed to confirm that the prototype fulfills both its functional and non-functional requirements.

6.0.1 Testing Methodology for Blockchain Applications

In this section, a range of tests designed to scrutinize both the functional and non-functional aspects of the blockchain implementation is presented REFFF:

- Chaincode testing: Smart contract testing is mainly about performing detailed functional testing of business logic
- Infrastructure testing: This type of testing verifies whether blockchain components can communicate and function with each other and if its various components are operating as expected
- Functional Testing: Validates the Blockchain, its subsystems and the associated business processes:
  - Network setup
  - Node creation
  - Transaction invocation
  - The consensus among nodes
  - Validation of the expected behavior
- Integration Testing: Integration tests are meant to verify the integration of nodes with the outside system, validating:
  - The communication between nodes in the blockchain
  - The message content
  - Integration with third-party applications and APIs
- Security Testing: A Blockchain network is only secure as its infrastructure. Security validation should be performed at a transaction and network level for Blockchain networks, including:

  - Identity/Access testing for users and administrators
  - Data integrity using tests that assess the ease of modifying data and assess its impact on application behavior
  - Perform encryption security and safety standards tests to prevent any misuse and misappropriation.

- Performance Testing: These are meant to verify the responsiveness of the system. This type of test enables the identification of performance bottlenecks and allows the definition of a metric to adjust the system

6.0.2 Performed Tests

First, unit tests were developed for the chaincode.

The contract unit tests were made to verify:

- The ledger Exclusivity Contract loading method.
- The process of creating, accessing, and updating specific information within an Exclusivity Contract.

All tests have a success and failure scenario. In Figure 6.1, it's possible to see an example of a test to the CreateAsset function. In the failure scenario, one parameter is missing in the request.

```
describe('Test CreateAsset', () => {
    it('should return error on CreateAsset', async () => {
        chaincodeStub.putState.rejects('failed inserting key');

        let contractManagement = new contractManagementContract();
        try {
            await contractManagement.CreateAsset(transactionContext, asset.ID, asset.PropertyId, asset.StartDate,
                asset.TerminationDate, asset.Terms, asset.Status);
            assert.fail('CreateAsset should have failed');
        } catch(err) {
            expect(err.name).to.equal('failed inserting key');
        }
    });

    it('should return success on CreateAsset', async () => {
        let contractManagement = new contractManagementContract();

        await contractManagement.CreateAsset(transactionContext, asset.ID, asset.BrokerId, asset.PropertyId, asset.StartDate,
            asset.TerminationDate, asset.Terms, asset.Status);

        let ret = JSON.parse((await chaincodeStub.getState(asset.ID)).toString());
        expect(ret).to.eql(asset);
    });
});
```

Figure 6.1: Create asset test

These unit tests validate the business logic and processes of the network, covering the smart contract testing from the previous section. These tests also validate the expected behavior of the nodes, validating all the conditions of it.

Based on the research conducted for this project, it was discovered that the methodology for conducting automated integration tests in Hyperledger Fabric is partially documented on the official Hyperledger Fabric GitHub repository. Hyperledger 2023a, however, it's only available using the Ginkgo, a testing framework exclusive for Golang language.

For that reason, a manual test of the system was performed locally, to test the blockchain core end-to-end and evaluate its responsiveness. By utilizing the same network configuration established in the PoC and Node.js API, a transaction to create an Exclusivity Contract object was conducted. The results showed that the asset was created successfully and the instance is visible in CouchDB database that is connected to both peers (Figure 6.2) that are part of the channel in which the transaction was conducted. Due to this mls-channel being shared by both organizations (Agency1 and Agency2), both can verify this asset in the respective peer database.

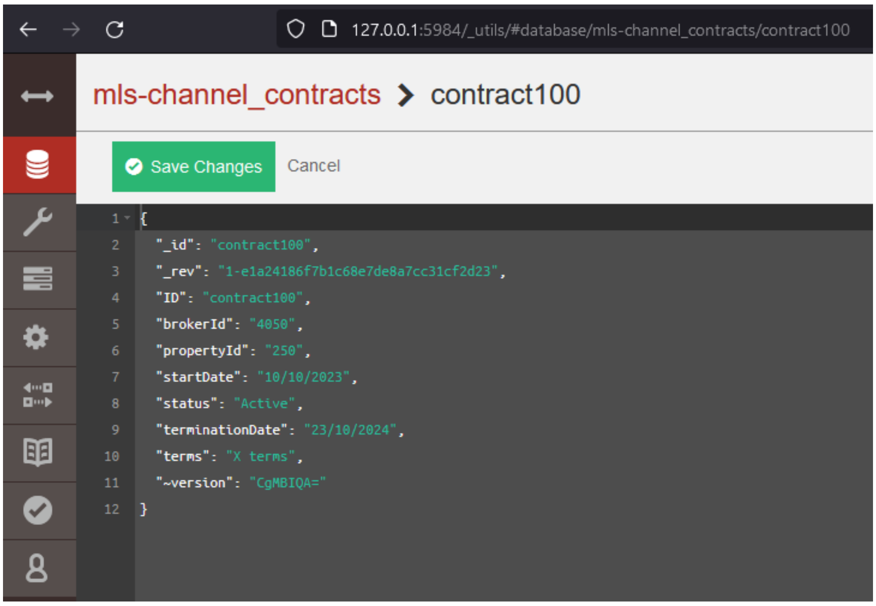

Figure 6.2: Exclusivity Contract instance in CouchDB

For the security testing, an attempt to tampering the node data was performed. Specifically, a request was sent from Agency2 to the update asset method, attempting to change the 'brokerID' field from '4050' to '5555'. By utilizing the Hyperledger Explorer and navigating to the transactions tab, a new transaction becomes visible. Upon inspecting the transaction details (Figure 6.3), it becomes apparent that a specific attribute within the displayed contract instance has been modified.

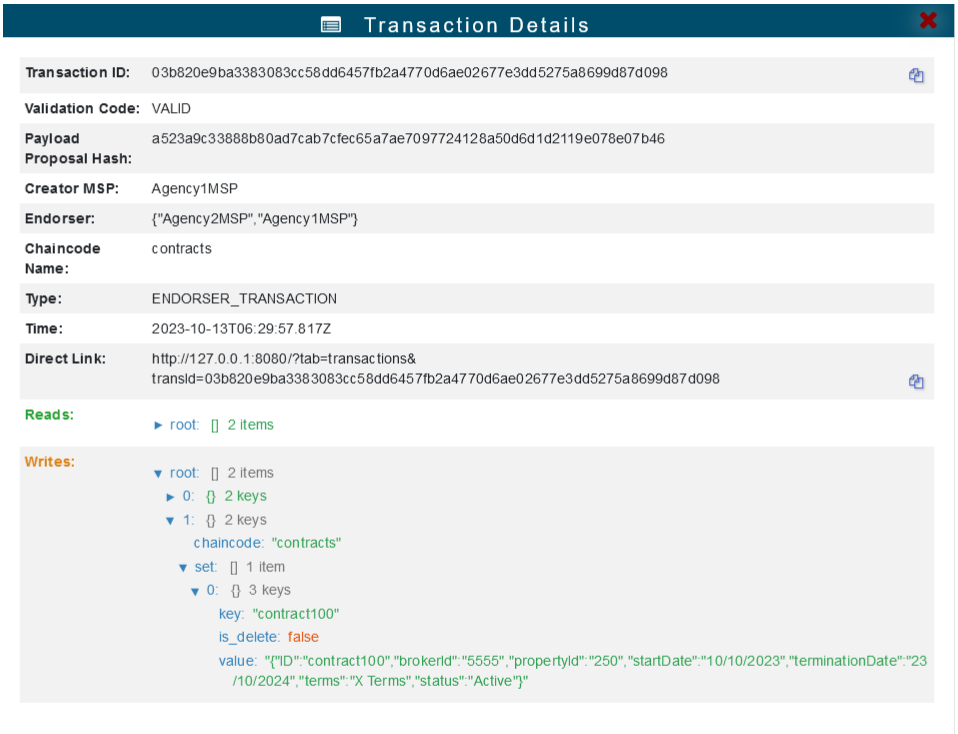

Figure 6.3: Hyperledger Explorer transaction details

Performance testing was not explored otherwise Hyperledger Caliper (Toumia, Berger, and Reiser 2021) would be a good option for that purpose.

6.0.3 Solution Evaluation

To evaluate the goals of this proof-of-concept application, the performed tests, presented in the previous section were crucial to demonstrating the results achieved through unit, integration and functional tests.

The application's privacy, transaction validity, and ledger auditability were considered and tested to ensure the standards of security and reliability.

The results indicated a positive outcome, showcasing the application's ability for nodes to communicate independently, eliminating the necessity for a central authority to serve as an intermediary.

Decentralizing data offers enhanced security compared to centralized solutions. This heightened security arises from the absence of a single point of failure; no single machine holds all the network's data records. This distributed setup effectively delays the potential success of attacks. The privacy concern wasn't completely illustrated due to limited testing – the implementation involved only one channel between two organizations. To comprehensively assess its capabilities, at least three organizations were necessary.

In summary, through this proof-of-concept, it becomes viable to redistribute the central authority's responsibilities among involved parties. This redistribution not only enhances security, privacy, and availability but also optimizes the overall operational structure.

Chapter 7

Conclusion

The conclusions chapter serves as the culmination of this in-depth exploration into the application of blockchain technology as a safeguard for data integrity within the context of real estate-oriented businesses, most specifically exclusivity contracts. Furthermore, this chapter also outlines the improvements that can be made to this project.

7.1 Project Synopsis

This project aims to bring value to the real estate sector by using the existing properties of distributed ledger technology to develop a proof-of-concept application that can help real estate agencies protect sensitive data. Moreover, it offers a glimpse into the technology's potential to create innovative business models. In this environment, former competitors could securely share client and property information, fostering collaboration and driving industry advancements.

During the project's initial phase, extensive research and data collection were conducted to select the most suitable distributed ledger technology and platform. Also, daily contact with DevScope was essential, to gain valuable insights into the project's business aspects, the exclusivity contract model and various rules.

Initially, the author had a different perspective on the project, considering the development of a comprehensive blockchain-based MLS (Multiple Listing Service) solution instead of concentrating solely on the exclusivity contract aspect. However, after discussions with DevScope, it became apparent that the MLS solution was overly complex and didn't directly address the core issue: ensuring the integrity of exclusivity contract data.

The subsequent task involved selecting the most appropriate platform from the array of options available in the market. This decision posed a considerable challenge, as it necessitated the careful consideration of numerous factors. After an in-depth analysis detailed in Chapter 4, the choice ultimately landed on Hyperledger Fabric.

The final stage of this project involves creating the Hyperledger Fabric application. This phase demonstrates the practical application of the chosen distributed ledger technology in the real estate sector. Specifically, it showcases how this technology facilitates the sharing of data related to exclusivity contracts without requiring intermediaries.

7.2 Contributions

This research contributes to the field of real estate and blockchain technology. The contributions of this thesis can be outlined as follows:

- Development of a Hyperledger Fabric Solution: This study presents a practical implementation of a Hyperledger Fabric-based system tailored for managing real estate exclusivity contracts. The developed solution showcases the feasibility and effectiveness of blockchain technology in enhancing the integrity and security of such contracts.
- In-Depth Analysis and Evaluation: Through extensive research and analysis, this thesis provides a comprehensive evaluation of other distributed ledger technologies and platforms available in the market. The assessment conducted in Chapter 4 aids future researchers and practitioners in understanding the selection criteria for similar projects.
- Addressing Industry-Specific Challenges: By focusing on the real estate sector and the intricacies of exclusivity contracts, this research addresses industry-specific challenges. The approach ensures that the developed solution is not only technically sound but also aligned with the requirements of the real estate domain. The integration with Kubernetes offers an adaptable solution capable of meeting the dynamic demands of the industry.
- Enhanced Security and Integrity: The proposed Hyperledger Fabric solution emphasizes paramount security, data integrity, and scalability. Through the integration with Kubernetes, the system achieves enhanced scalability and resilience, ensuring seamless operation even under high transaction loads.
- Contributions to Academic and Professional Communities: This thesis contributes to the academic community by offering valuable insights into the intersection of real estate management and blockchain technology. Additionally, it serves as a resource for professionals in the real estate industry, providing a foundation for implementing secure, transparent, and efficient contract management systems.

  The code for the Proof of Concept application can be found in a public repository in Github:

  `https://github.com/danielfmdias/Blockchain-and-Property-Technology.git`

7.3 Future Work

As for future work, this proof-of-concept application code can be improved, by adding some new functionalities such as:

1. Expand the number of channels and organizations to showcase the privacy capabilities enabled by Hyperledger Fabric.
2. Introduce a broker tasked with capturing events from a queue linked to actions and events carried out by real estate agents within the Multiple Listing Service. This approach ensures a streamlined integration process for blockchain technology, making it exceptionally accessible and easy to adopt.
3. Implementation of GDPR Compliance Measures.

Bibliography

Ahmad, Iftikhar et al. (2021). "Real Estate Management via a Decentralized Blockchain Platform". en. In: *Computers, Materials & Continua* 66.2, pp. 1813–1822. issn: 1546-2226. doi: 10.32604/cmc.2020.013048. url: https://www.techscience.com/cmc/v66n2/40660 (visited on 11/07/2022).

Androulaki, Elli et al. (2018). "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". en. In: url: https://dl.acm.org/doi/pdf/10.1145/3190508.3190538.

Atzori, Marcella (Mar. 2017). "Blockchain technology and decentralized governance: Is the state still necessary?" In: *Journal of Governance and Regulation* 6. doi: 10.22495/jgr_v6_i1_p5. url: http://dx.doi.org/10.22495/jgr_v6_i1_p5.

Auinger, Andreas and René Riedl (2018). "Blockchain and Trust: Refuting Some Widely-held Misconceptions". en. In: url: https://www.researchgate.net/publication/328465252_Blockchain_and_Trust_Refuting_Some_Widely-held_Misconceptions.

Baird, Dr Leemon, Mance Harmon, and Paul Madsen (2020). "Hedera: A Public Hashgraph Network & Governing Council". en. In: url: https://www.securities.io/wp-content/uploads/2022/05/hh_whitepaper_v2.1-20200815.pdf.

Baird, Leemon (2016). "The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance". en. In: url: https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf.

Baird, Leemon and Atul Luykx (Aug. 2020). "The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers". en. In: *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. Barcelona, Spain: IEEE, pp. 1–7. isbn: 978-1-72816-371-0. doi: 10.1109/COINS49042.2020.9191430. url: https://ieeexplore.ieee.org/document/9191430/ (visited on 01/16/2023).

Bashir, Imran (Aug. 2020). *Mastering Blockchain 3rd Edition*. isbn: 978-1-83921-319-9.

Bayer, Dave, Stuart Haber, and W. Stornetta (Sept. 1999). "Improving the Efficiency and Reliability of Digital Time-Stamping". In: issn: 978-1-4613-9325-2. doi: 10.1007/978-1-4613-9323-8_24.

Benčić, Federico Matteo, Pavle Skočir, and Ivana Podnar Žarko (2019). "DL-Tags: DLT and Smart Tags for Decentralized, Privacy-Preserving, and Verifiable Supply Chain Management". In: *IEEE Access* 7. Conference Name: IEEE Access, pp. 46198–46209. issn: 2169-3536. doi: 10.1109/ACCESS.2019.2909170.

Bhandary, Mohan, Manish Parmar, and Dayanand Ambawade (June 2020). "A Blockchain Solution based on Directed Acyclic Graph for IoT Data Security using IoTA Tangle". In: *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 827–832. doi: 10.1109/ICCES48766.2020.9137858.

Bishr, Aisha Bin (Jan. 2019). "Dubai: A City Powered by Blockchain". en. In: *Innovations: Technology, Governance, Globalization* 12.3-4, pp. 4–8. issn: 1558-2477, 1558-2485. doi: 10.1162/inov_a_00271. url: https://direct.mit.edu/itgg/article/12/3-4/4-8/9846 (visited on 01/31/2023).

Brogan, James, Immanuel Baskaran, and Navin Ramachandran (Jan. 2018). "Authenticating Health Activity Data Using Distributed Ledger Technologies". en. In: *Computational and Structural Biotechnology Journal* 16, pp. 257–266. issn: 2001-0370. doi: 10.1016/j.csbj.2018.06.004. url: https://www.sciencedirect.com/science/article/pii/S2001037018300345 (visited on 02/08/2023).

Brummer, Chris (Mar. 2022). *Disclosure, Dapps and DeFi*. en. SSRN Scholarly Paper. Rochester, NY. url: https://papers.ssrn.com/abstract=4065143 (visited on 02/05/2023).

Budman, Matthew et al. (2021). "Deloitte 2021 Blockchain survey". en. In: url: https://www2.deloitte.com/us/en/insights/topics/understanding-blockchain-potential/global-blockchain-survey.html.

Burkhardt, Daniel, Maximilian Werling, and Heiner Lasi (June 2018). "Distributed Ledger". en. In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. Stuttgart: IEEE, pp. 1–9. isbn: 978-1-5386-1469-3. doi: 10.1109/ICE.2018.8436299. url: https://ieeexplore.ieee.org/document/8436299/ (visited on 12/31/2022).

Butcher, Mike (Oct. 2020). *Propy, a blockchain-verified platform for selling houses, raises funding from Tim Draper*. en-US. url: https://techcrunch.com/2020/10/02/propy-a-blockchain-verified-platform-for-selling-houses-raises-funding-from-tim-draper/ (visited on 02/24/2023).

– (May 2021). *Blockchain startup Propy plans first-ever auction of a real apartment as a collectible NFT*. en-US. url: https://techcrunch.com/2021/05/25/blockchain-startup-propy-plans-first-ever-auction-of-a-real-apartment-as-a-collectible-nft/ (visited on 02/24/2023).

Buterin, Vitalik (2014). *DAOs, DACs, DAs and More: An Incomplete Terminology Guide*. en. url: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide (visited on 02/24/2023).

Calma, Justine (Sept. 2022). *How The Merge will slash Ethereum's climate pollution*. en-US. url: https://www.theverge.com/2022/9/6/23339594/ethereum-merge-energy-pollution-proof-of-stake (visited on 02/09/2023).

Cao, Shuyan et al. (2017). "A Review of Researches on Blockchain". en. In: *Digital Innovation*. url: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1012&context=whiceb2017.

Castro, Miguel and Barbara Liskov (2019). "Practical Byzantine Fault Tolerance". en. In.

Chaum, D.L. (1982). *Computer Systems Established, Maintained and Trusted by Mutually Suspicious Groups*. University of California, Berkeley. url: https://books.google.pt/books?id=C8xQnQEACAAJ.

Chen, James (2022). *Multiple Listing Service (MLS): Definition, Benefits, and Fees*. en. url: https://www.investopedia.com/terms/m/multiple-listing-service-mls.asp (visited on 12/07/2022).

Chowdhury, Mohammad Jabed Morshed et al. (2019). "A Comparative Analysis of Distributed Ledger Technology Platforms". In: *IEEE Access* 7. Conference Name: IEEE Access, pp. 167930–167943. issn: 2169-3536. doi: 10.1109/ACCESS.2019.2953729.

Chuen, David (Jan. 2015). *Handbook of Digital Currency*. Academic Press. isbn: 978-0128021170.

Collindres, Jorge Constantino, Matt Regan, and Guillermo Pena Panting (2016). "Using Blockchain to Secure Honduran Land Titles". en. In: url: https://s3.amazonaws.com/ipri2016/casestudy_collindres.pdf.

Dreyfuss-Chavez, Gertrude (June 2016). *Sweden tests blockchain technology for land registry*. en-US. url: http://finance.yahoo.com/news/sweden-tests-blockchain-technology-land-registry-201944959--finance.html (visited on 01/31/2023).

DuPont, Quinn (Nov. 2017). "A history and ethnography of "The DAO," a failed decentralized autonomous organization". en. In: *Bitcoin and Beyond*. 1st ed. London: Routledge, pp. 157–177. isbn: 978-1-315-21190-9. doi: 10.4324/9781315211909-8. url: https://www.taylorfrancis.com/books/9781315211909/chapters/10.4324/9781315211909-8 (visited on 11/15/2022).

Duranti, Luciana and Corinne Rogers (Jan. 2014). "Trust in Online Records and Data". In: pp. 203–216. url: https://www.researchgate.net/publication/290042093_Trust_in_Online_Records_and_Data.

Dyer, Ian (Dec. 2017). *IOTA - An Innovative Cryptocurrency for the Revolutionary Internet of Things*. en-US. url: https://banyanhill.com/iota-cryptocurrency-iot/ (visited on 02/08/2023).

EconoTimes (Sept. 2016). *Bitland partners with CCEDK to improve blockchain land registry in West Africa*. en. url: https://www.econotimes.com/Bitland-partners-with-CCEDK-to-improve-blockchainland-registry-in-West-Africa-271517 (visited on 02/11/2023).

Farghaly, Mahmoud et al. (2019). *Annual report: Dubai real estate sector performance 2019*. en. Tech. rep. Dubai Land Department. url: https://dubailand.gov.ae/media/qq0nxuoi/annual-report-real-estate-sector-performance-2019.pdf.

Fernández-Caramés, Tiago M. and Paula Fraga-Lamas (2018). "A Review on the Use of Blockchain for the Internet of Things". In: *IEEE Access* 6. Conference Name: IEEE Access, pp. 32979–33001. issn: 2169-3536. doi: 10.1109/ACCESS.2018.2842685.

Frankenfield, Jake (2021). *What Is Tangle?* en. url: https://www.investopedia.com/terms/t/tangle-cryptocurrency.asp (visited on 02/08/2023).

Franks, Patricia (Feb. 2020). "Implications of blockchain distributed ledger technology for records management and information governance programs". In: *Records Management Journal* ahead-of-print. doi: 10.1108/RMJ-08-2019-0047.

Fridgen, Gilbert et al. (2017). *Unchaining Social Businesses–Blockchain as the Basic Technology of a Crowdlending Platform*. url: http://www.fim-rc.de/Paperbibliothek/Veroeffentlicht/712/wi-712.pdf.

Gayialis, Sotiris, Evripidis Kechagias, Grigorios D. Konstantakopoulos, et al. (Aug. 2021). "An Approach for Creating a Blockchain Platform for Labeling and Tracing Wines and Spirits". In: pp. 81–89. isbn: 978-3-030-85909-1. doi: 10.1007/978-3-030-85910-7_9.

Gayialis, Sotiris, Evripidis Kechagias, Georgios Papadopoulos, et al. (Aug. 2019). "Design of a Blockchain-Driven System for Product Counterfeiting Restraint in the Supply Chain". In: pp. 474–481. isbn: 978-3-030-29999-6. doi: 10.1007/978-3-030-30000-5_59.

Geroni, Diego (Jan. 2021). *Hybrid Blockchain: The Best of Both Worlds*. en-US. url: https://101blockchains.com/hybrid-blockchain/ (visited on 02/09/2023).

Greenspan, Dr Gideon (2015). *MultiChain Private Blockchain — White Paper*. url: https://www.multichain.com/download/MultiChain-White-Paper.pdf.

Gupta, Ashutosh et al. (2020). "Tokenization of Real Estate Using Blockchain Technology". en. In: *Applied Cryptography and Network Security Workshops*. Ed. by Jianying Zhou et al. Vol. 12418. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 77–90. doi: 10.1007/978-3-030-61638-0_5. url: http://link.springer.com/10.1007/978-3-030-61638-0_5 (visited on 01/30/2023).

Haber, Stuart and W. Scott Stornetta (Jan. 1991). "How to time-stamp a digital document". en. In: *Journal of Cryptology* 3.2, pp. 99–111. issn: 1432-1378. doi: 10.1007/BF00196791. url: https://doi.org/10.1007/BF00196791 (visited on 02/06/2023).

Hawlitschek, Florian, Benedikt Notheisen, and Timm Teubner (May 2018). "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy". en. In: *Electronic Commerce Research and Applications* 29, pp. 50–63. issn: 1567-4223. doi: 10.1016/j.elerap.2018.03.005. url: https://www.sciencedirect.com/science/article/pii/S1567422318300292 (visited on 02/07/2023).

Hearn, Mike and Richard Gendal Brown (2019). "Corda: A distributed ledger". en. In: p. 73. url: https://docs.r3.com/en/pdf/corda-technical-whitepaper.pdf.

Hedera (2020). *Hashgraph Platform is Now Available as Open Review*. en. url: https://hedera.com/blog/hashgraph-platform-is-now-available-as-open-review (visited on 02/08/2023).

Holbrook, Morris B. (1996). "Special Session Summary Customer Value C a Framework For Analysis and Research". en. In: *ACR North American Advances* NA-23. url: https://www.acrwebsite.org/volumes/7929/volumes/v23/NA-23/full (visited on 02/16/2023).

Honduras to build land title registry using bitcoin technology (May 2015). en-US. url: http://news.yahoo.com/honduras-build-land-title-registry-using-bitcoin-technology-162701917.html (visited on 01/31/2023).

Hoxha, Ledina (2018). "Hashgraph the Future of Decentralized Technology and the End of Blockchain". en. In: *European Journal of Engineering and Formal Sciences* 2.1. doi: 10.26417/ejef.v2i2.p86-89.

Hwang, Ching-Lai and Kwangsun Yoon (1981). "Multiple Attribute Decision Making: Methods and Applications - A State-of-the-Art Survey". In: *Lecture notes in economics and mathematical systems*.

Hyperledger (2023a). *fabric*. https://github.com/hyperledger/fabric/tree/main/integration.

– (2023b). *fabric-sdk-node*. https://github.com/hyperledger/fabric-sdk-node.

Ismail, Leila and Huned Materwala (Oct. 2019). "A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions". en. In: *Symmetry* 11.10. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 1198. issn: 2073-8994. doi: 10.3390/sym11101198. url: https://www.mdpi.com/2073-8994/11/10/1198 (visited on 01/19/2023).

Jha, Prashant (2022). *The aftermath of Axie Infinity's $650M Ronin Bridge hack*. url: https://cointelegraph.com/news/the-aftermath-of-axie-infinity-s-650m-ronin-bridge-hack (visited on 02/11/2023).

Koen, Peter (Nov. 2007). "The Fuzzy Front End for Incremental, Platform, and Breakthrough Products". In: pp. 81–91. isbn: 9780470172483. doi: 10.1002/9780470172483.ch6.

Kshetri, Nir and Jeffrey Voas (Mar. 2018). "Blockchain in Developing Countries". en. In: *IT Professional* 20.2, pp. 11–14. issn: 1520-9202, 1941-045X. doi: 10.1109/MITP.2018.021921645. url: https://ieeexplore.ieee.org/document/8338009/ (visited on 01/31/2023).

Labs, Casper (2023). *The State of Enterprise Blockchain Adoption in 2023*. en. url: https://casperlabs.io/blog/the-state-of-enterprise-blockchain-adoption-in-2023 (visited on 02/12/2023).

Lai, Roy and David LEE Kuo Chuen (2018). "Chapter 7 - Blockchain – From Public to Private". In: *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*. Ed. by

David Lee Kuo Chuen and Robert Deng. Academic Press, pp. 145–177. isbn: 978-0-12-812282-2. doi: https://doi.org/10.1016/B978-0-12-812282-2.00007-3. url: https://www.sciencedirect.com/science/article/pii/B9780128122822000073.

Lazuashvili, Nino (May 2019). "Integration of the Blockchain Technology Into The Land Registration System. A Case Study of Georgia". PhD thesis. doi: 10.13140/RG.2.2.35689.13920/1.

Lenz, Rainer (2019). "Managing Distributed Ledgers: Blockchain and Beyond". en. In: *SSRN Electronic Journal*. issn: 1556-5068. doi: 10.2139/ssrn.3360655. url: https://www.ssrn.com/abstract=3360655 (visited on 12/10/2022).

Locke, Taylor (2023). *Ethereum's Russian-born cofounder has been quietly supporting a DAO that raised $8 million in crypto for Ukraine. His dad is even more involved.* en. url: https://fortune.com/crypto/2022/03/31/ukraine-dao-crypto-donations-vitalik-buterin/ (visited on 02/11/2023).

Mação, Ana (2022). *Contrato de Exclusividade*. pt-pt. Text. Publisher: Ana Mação - KW. url: https://ana-macao-kw.pt/contrato-de-exclusividade (visited on 11/14/2022).

Masse, Bryson (2017). *Apartment purchased with Ethereum raises hopes, questions*. en-US. url: https://www.inman.com/2017/09/29/apartment-purchased-with-ethereum-raises-hopes-questions/ (visited on 02/24/2023).

Massias, X. S. Avila, and J.-J. QuisquaterUCL (1999). "DESIGN OF A SECURE TIMESTAMPING SERVICEWITH MINIMAL TRUST REQUIREMENTH". In: url: https://www.semanticscholar.org/paper/DESIGN-OF-A-SECURE-TIMESTAMPING-SERVICEWITH-MINIMAL-Massias-Avila/7fd3174853c4530fba40e0f064870d2b5bf9a979 (visited on 02/08/2023).

Mayer, Emma (Aug. 2021). *$611 Million in Cryptocurrencies Stolen in Massive Hack*. en. Section: News. url: https://www.newsweek.com/611-million-cryptocurrencies-stolen-massive-hack-1617999 (visited on 02/11/2023).

McMurren, Juliet, Andrew Young, and Stefaan Verhulst (2018). "Addressing Transaction Costs Through Blockchain and Identity in Swedish Land Transfers". en. In: url: https://blockchan.ge/blockchange-appendices.pdf.

Mehendale, Deepashree K, Reshma S Masurekar, and Harsha V Patil (2019). "Implications of Block Chain in Real Estate Industry". en. In: 8.1, p. 4. url: https://www.ijrte.org/wp-content/uploads/papers/v8i1/A5388058119.pdf.

Melo, Carlos et al. (June 2018). "Dependability Evaluation of a Blockchain-as-a-Service Environment". In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. ISSN: 1530-1346, pp. 00909–00914. doi: 10.1109/ISCC.2018.8538752.

Miles, L.D. (2015). *Techniques of Value Analysis and Engineering*. McGraw-Hill. isbn: 978-0-07-041926-1. url: https://books.google.pt/books?id=XxhbDwAAQBAJ.

Miles, Larry, Su Mi Dahlgaard-Park, and Warwick Manufacturing Group (2007). *QFD: The Integration of Quality Function Deployment and Six Sigma*. Gower Publishing, Ltd.

Mintah, Kwabena et al. (May 2020). "Skin lands in Ghana and application of blockchain technology for acquisition and title registration". In: *Journal of Property, Planning and Environmental Law* ahead-of-print. doi: 10.1108/JPPEL-12-2019-0062.

Mondal, Saikat et al. (June 2019). "Blockchain Inspired RFID-Based Information Architecture for Food Supply Chain". In: *IEEE Internet of Things Journal* 6.3. Conference Name: IEEE Internet of Things Journal, pp. 5803–5813. issn: 2327-4662. doi: 10.1109/JIOT.2019.2907658.

Moskov, Alex (Mar. 2020). *Ethereum Classic vs Ethereum: What's the Difference?* en-US. Section: Articles. url: https://coincentral.com/ethereum-classic-vs-ethereum/ (visited on 02/05/2023).

Moura, Tiago Augusto Pinto (2020). "Design patterns for Ethereum smart contracts". por. Accepted: 2020-11-05T15:33:11Z. MA thesis. url: https://recipp.ipp.pt/handle/10400.22/16441 (visited on 01/19/2023).

Nakamoto, Satoshi (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System". en. In: p. 9.

Natarajan, H., S. Krause, and Helen Luskin Gradstein (2017). "Distributed Ledger Technology (DLT) and blockchain : Fintech note no. 1". In: url: https://www.semanticscholar.org/paper/Distributed-Ledger-Technology-(DLT)-and-blockchain-Natarajan-Krause/0188b89179645d52e4a96f8d1054124672e2d59e (visited on 02/06/2023).

Norton, Kaiti (Sept. 2021). *What is DocuSign?* en-US. url: https://www.webopedia.com/definitions/docusign/ (visited on 02/24/2023).

Notheisen, Benedikt, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam (Dec. 2017). "Trading Real-World Assets on Blockchain". en. In: *Business & Information Systems Engineering* 59.6, pp. 425–440. issn: 1867-0202. doi: 10.1007/s12599-017-0499-8. url: https://doi.org/10.1007/s12599-017-0499-8 (visited on 02/09/2023).

Novo, Oscar (Apr. 2018). "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT". In: *IEEE Internet of Things Journal* 5.2. Conference Name: IEEE Internet of Things Journal, pp. 1184–1195. issn: 2327-4662. doi: 10.1109/JIOT.2018.2812239.

NRI (2016). *Survey on Blockchain Technologies and Related Services - Nomura Research Institute*. url: https://smallake.kr/wp-content/uploads/2016/06/0531_01f.pdf.

O'Neill, Joseph, Matt Hussey, and Scott Chipolina (2022). *What Are Decentralized Applications (Dapps)? - Decrypt*. url: https://decrypt.co/resources/what-are-decentralized-applications-dapps (visited on 02/11/2023).

Ølnes, Svein, Jolien Ubacht, and Marijn Janssen (Sept. 2017). "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing". en. In: *Government Information Quarterly* 34.3, pp. 355–364. issn: 0740-624X. doi: 10.1016/j.giq.2017.09.007. url: https://www.sciencedirect.com/science/article/pii/S0740624X17303155 (visited on 02/08/2023).

Ongaro, Diego and John Ousterhout (2014). "In Search of an Understandable Consensus Algorithm". en. In: pp. 305–319. isbn: 978-1-931971-10-2. url: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro (visited on 02/09/2023).

Osterwalder, A. et al. (2015). *Value Proposition Design: How to Create Products and Services Customers Want*. The Strategyzer Series. Wiley. isbn: 978-1-118-96806-2. url: https://books.google.pt/books?id=jgu5BAAAQBAJ.

Paul, Jean (Feb. 2020). *Blockchain explained and its application to payments*. en-GB. url: https://www.paiementor.com/blockchain-explained-application-payments/ (visited on 02/08/2023).

Peffers, Ken et al. (Dec. 2007). "A Design Science Research Methodology for Information Systems Research". en. In: *Journal of Management Information Systems* 24.3, pp. 45–77. issn: 0742-1222, 1557-928X. doi: 10.2753/MIS0742-1222240302. url: https://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302 (visited on 02/06/2023).

Polge, Julien, Jérémy Robert, and Yves Le Traon (June 2021). "Permissioned blockchain frameworks in the industry: A comparison". en. In: *ICT Express* 7.2, pp. 229–233. issn: 2405-9595. doi: 10.1016/j.icte.2020.09.002. url: https://www.sciencedirect.com/science/article/pii/S2405959520301909 (visited on 11/29/2022).

Popov, Serguei (2018). "The tangle". In: *White paper* 1.3, p. 30. url: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf.

Porter, Michael E (1990). "The Competitive Advantage of Nations". In: *Harvard Business Review* 68.2.

Prospera (2022). *Próspera Announces Bitcoin Bond Authority in the First...* en. url: https://prospera.hn/news/press-releases/pr%C3%B3spera-announces-bitcoin-bond-authority-in-the-first-crypto-friendly-jurisdiction-that-is-fully-aml-kyc-compliant (visited on 02/11/2023).

Al-Quraan, Mohammad et al. (Nov. 2021). *Edge-Native Intelligence for 6G Communications Driven by Federated Learning: A Survey of Trends and Challenges*. en. arXiv:2111.07392 [cs]. url: http://arxiv.org/abs/2111.07392 (visited on 02/08/2023).

Rauchs, Michel et al. (Aug. 2018). *Distributed Ledger Technology Systems: A Conceptual Framework*. en. SSRN Scholarly Paper. Rochester, NY. doi: 10.2139/ssrn.3230013. url: https://papers.ssrn.com/abstract=3230013 (visited on 12/09/2022).

Raval, Siraj (Aug. 2016). "Decentralized Applications: Harnessing Bitcoin's Blockchain Technology". In: url: https://www.semanticscholar.org/paper/Decentralized-Applications%3A-Harnessing-Bitcoin%27s-Raval/f473891a959736e8cb9304c0394c1030c0b65455 (visited on 02/05/2023).

Rensin, David K. (June 2015). *Kubernetes - scheduling the future at Cloud Scale*. url: https://research.google/pubs/pub43826/.

Rodima-Taylor, Daivi (June 2021). "Digitalizing land administration: The geographies and temporalities of infrastructural promise". en. In: *Geoforum* 122, pp. 140–151. issn: 0016-7185. doi: 10.1016/j.geoforum.2021.04.003. url: https://www.sciencedirect.com/science/article/pii/S0016718521001020 (visited on 01/31/2023).

Rofiq, Faiz and Im Rafid (Aug. 2021). *ImBlock: A Blockchain Implementation of Land Ownership Recording System using Hyperledger Fabric*. doi: 10.13140/RG.2.2.15654.45123.

Sazandrishvili, George (2020). "Asset tokenization in plain English". en. In: *Journal of Corporate Accounting & Finance* 31.2. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcaf.22432, pp. 68–73. issn: 1097-0053. doi: 10.1002/jcaf.22432. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcaf.22432 (visited on 02/26/2023).

Schechter, Laurence (1984). "A Normative Conception of Value". In: *Progressive Grocer, Executive Report*.

Shabandri, Bilal and Piyush Maheshwari (Mar. 2019). "Enhancing IoT Security and Privacy Using Distributed Ledgers with IOTA and the Tangle". In: *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 1069–1075. doi: 10.1109/SPIN.2019.8711591.

Shang, Qiuyun and Allison Price (2018). "A Blockchain-based land titling project in the Republic of Georgia". en. In: 12.3. url: http://www.mitpressjournals.org/doi/pdf/10.1162/inov_a_00276.

Sharma, Rakesh (2021). *Bit Gold*. en. url: https://www.investopedia.com/terms/b/bit-gold.asp (visited on 02/11/2023).

– (2022). *What Is Decentralized Finance (DeFi) and How Does It Work?* en. url: https://www.investopedia.com/decentralized-finance-defi-5113835 (visited on 02/24/2023).

ShelterZoom (Nov. 2021). *ShelterZoom Launches "1REport", New Turnkey Blockchain-Based Real Estate Listing and Marketing Solution*. en-US. url: https://www.yahoo.

com/now/shelterzoom-launches-1report-turnkey-blockchain-162000659.html (visited on 02/24/2023).

Shen, Charles and Feniosky Pena-Mora (2018). “Blockchain for Cities—A Systematic Literature Review”. In: *IEEE Access* 6, pp. 76787–76819. issn: 2169-3536. doi: 10.1109/ACCESS.2018.2880744. url: https://ieeexplore.ieee.org/document/8531608/ (visited on 02/08/2023).

Shrestha, Rakesh and Seung Yeob Nam (2019). “Regional Blockchain for Vehicular Networks to Prevent 51% Attacks”. en. In: *IEEE Access* 7, pp. 95033–95045. issn: 2169-3536. doi: 10.1109/ACCESS.2019.2928753. url: https://ieeexplore.ieee.org/document/8763943/ (visited on 02/08/2023).

Shuttleworth, David (2021). *What Is A DAO And How Do They Work?* en. url: https://consensys.net/blog/blockchain-explained/what-is-a-dao-and-how-do-they-work/ (visited on 02/05/2023).

Siira, Erkki et al. (June 2017). *Designing and Implementing Common Market for Cross-Game Purchases between Mobile Games*. doi: 10.18690/978-961-286-043-1.37.

Sudin, M.N. et al. (Apr. 2017). “AHP method and application example for the robust Multi-Criteria design concept selection”. In: *ARPN journal of engineering and applied science* 12.

Szabo, Nick (1996). *Smart Contracts: Building Blocks for Digital Markets*. url: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (visited on 12/15/2022).

Takyar, Akash (Mar. 2018). *Hedera Hashgraph vs Blockchain | Comparison*. en-US. url: https://www.leewayhertz.com/hashgraph-vs-blockchain/ (visited on 02/08/2023).

Tapscott, Alex and Don Tapscott (Mar. 2017). “How Blockchain Is Changing Finance”. In: *Harvard Business Review*. Section: Financial markets. issn: 0017-8012. url: https://hbr.org/2017/03/how-blockchain-is-changing-finance (visited on 02/08/2023).

Taylor, Paul et al. (Feb. 2019). “A systematic literature review of blockchain cyber security”. In: *Digital Communications and Networks* 6. doi: 10.1016/j.dcan.2019.01.005.

Toumia, Sadok Ben, Christian Berger, and Hans; P. Reiser (2021). url: https://arxiv.org/pdf/2111.15399.pdf.

Ullah, Fahim and Fadi Al-Turjman (Feb. 2021). “A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities”. en. In: *Neural Computing and Applications*. issn: 0941-0643, 1433-3058. doi: 10.1007/s00521-021-05800-6. url: http://link.springer.com/10.1007/s00521-021-05800-6 (visited on 11/04/2022).

Wang, Qiang and Min Su (Aug. 2020). “Integrating blockchain technology into the energy sector — from theory of blockchain to research and application of energy blockchain”. en. In: *Computer Science Review* 37, p. 100275. issn: 1574-0137. doi: 10.1016/j.cosrev.2020.100275. url: https://www.sciencedirect.com/science/article/pii/S1574013720300241 (visited on 02/08/2023).

Wang, Qin et al. (Oct. 2021). *Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges*. en. arXiv:2105.07447 [cs]. url: http://arxiv.org/abs/2105.07447 (visited on 02/24/2023).

Wang, Shuai et al. (Dec. 2018). “Blockchain-Powered Parallel Healthcare Systems Based on the ACP Approach”. en. In: *IEEE Transactions on Computational Social Systems* 5.4, pp. 942–950. issn: 2329-924X, 2373-7476. doi: 10.1109/TCSS.2018.2865526. url: https://ieeexplore.ieee.org/document/8449329/ (visited on 02/08/2023).

What is Hedera? (2023). en. url: https://hedera.com/learning/hedera-hashgraph/what-is-hedera-hashgraph (visited on 02/08/2023).

wpadmin (Oct. 2017). *Swirlds and CULedger Collaborate to Deliver High Performance, Secure, Distributed Applications to Credit Unions*. en-US. url: https://www.swirlds.com/swirlds-culedger-collaborate-deliver-high-performance-secure-distributed-applications-credit-unions/ (visited on 02/08/2023).

Xiao, Yang et al. (Sept. 2020). "PrivacyGuard: Enforcing Private Data Usage Control with Blockchain and Attested Off-Chain Contract Execution". In: *Computer Security – ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, pp. 610–629. isbn: 978-3-030-59012-3. doi: 10.1007/978-3-030-59013-0_30. url: https://doi.org/10.1007/978-3-030-59013-0_30 (visited on 02/08/2023).

Xie, Junfeng et al. (2019). "A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges". en. In: *IEEE Communications Surveys & Tutorials* 21.3, pp. 2794–2830. issn: 1553-877X, 2373-745X. doi: 10.1109/COMST.2019.2899617. url: https://ieeexplore.ieee.org/document/8642861/ (visited on 02/08/2023).

Yeo, Geoffrey (Oct. 2013). "Trust and context in cyberspace". In: *Archives and Records* 34.2. Publisher: Routledge _eprint: https://doi.org/10.1080/23257962.2013.825207, pp. 214–234. issn: 2325-7962. doi: 10.1080/23257962.2013.825207. url: https://doi.org/10.1080/23257962.2013.825207 (visited on 02/09/2023).

Yewale, Aniket (2018). url: https://digitalscholarship.unlv.edu/cgi/viewcontent.cgi?article=4395&context=thesesdissertations.

Zahoor, Ehtesham (2023). *Kubernetes: A comprehensive tutorial for beginners*. url: https://www.educative.io/blog/kubernetes-tutorial-for-beginners.

Zapotochnyi, Andrew (Feb. 2022). *What is a Decentralised Autonomous Organization (DAO)?* en-Ca. url: https://blockgeeks.com/guides/what-is-a-decentralised-autonomous-organization-dao/ (visited on 02/05/2023).

Zeithaml, Valarie (July 1988). "Consumer Perceptions of Price, Quality and Value: A Means-End Model and Synthesis of Evidence". In: *Journal of Marketing* 52. doi: 10.1177/002224298805200302.

Zeng, Xueyun et al. (Aug. 2019). "A consortium blockchain paradigm on hyperledger-based peer-to-peer lending system". In: *China Communications* 16.8. Conference Name: China Communications, pp. 38–50. issn: 1673-5447. doi: 10.23919/JCC.2019.08.004.

Zhang, Hongwei et al. (Aug. 2022). "A Novel Distributed Ledger Technology Structure for Wireless Sensor Networks Based on IOTA Tangle". en. In: *Electronics* 11.15, p. 2403. issn: 2079-9292. doi: 10.3390/electronics11152403. url: https://www.mdpi.com/2079-9292/11/15/2403 (visited on 01/04/2023).

Zheng, Zibin, Shaoan Xie, Hong Ning Dai, et al. (2018). "Blockchain challenges and opportunities: a survey". en. In: *International Journal of Web and Grid Services* 14.4, p. 352. issn: 1741-1106, 1741-1114. doi: 10.1504/IJWGS.2018.095647. url: http://www.inderscience.com/link.php?id=95647 (visited on 01/30/2023).

Zheng, Zibin, Shaoan Xie, Hongning Dai, et al. (June 2017). "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". en. In: *2017 IEEE International Congress on Big Data (BigData Congress)*. Honolulu, HI, USA: IEEE, pp. 557–564. isbn: 978-1-5386-1996-4. doi: 10.1109/BigDataCongress.2017.85. url: http://ieeexplore.ieee.org/document/8029379/ (visited on 01/02/2023).