POLITECNICO DI TORINO

Master's Degree in Data Science And Engineering

Master's Degree Thesis

IoT and Blockchain Integration for Optimizing Smart City Public Transport: A Feasibility Case Study

Supervisor

Danilo BAZZANELLA

Candidate

Elisa FERAUD

JULY 2023

Summary

Nowadays the Internet of Things (IoT) is experiencing an important development in a variety of fields, leading to the rise of new applications useful for everyday life. In order to overcome the limitations of this technology, the integration of the Blockchain with IoT applications is the subject of several studies. In this scenario, TurinTech, an Italian company where I projected this thesis, is evaluating the feasibility of possible implementations of the Blockchain and IoT technologies in the Smart Cities and Automotive domains. The main purpose of this thesis is to provide a description of a possible application involving IoT and Blockchain technologies. Specifically, the proposed project aims to extract data from public vehicles, such as buses, in order to analyze them through IoT systems and store the results in the Blockchain, thus monitoring the quality of the drives and the fuel consumption. The objective is to encourage drivers to drive more carefully in order to reduce fuel consumption and to offer a better service to citizens: through a reward system, drivers receive a reward proportional to the quality of their drives, by monitoring data such as speed, engine RPM and the load of the vehicle. Thus, the two technologies are analyzed, focusing on their respective limitations. IoT devices, such as sensors, process a huge volume of data, such as personal information, thus security, data integrity, and data privacy are essential aspects to be taken into account. Centralized databases could lead to weak solutions, due to the presence of a single server, such as the Single-Point-Of-Failure. A distributed or decentralized solution could be better in terms of failure tolerance, efficiency, and computing power. Blockchain can be a possible storage solution, able to guarantee high levels of security and data integrity.
Thanks to the immutability of stored transactions, Blockchain allows the monitoring of historical data. Furthermore, costs related to a trusted third party are reduced. Despite Blockchain being born as a public system, several private platforms have been developed, in order to meet companies' requirements, such as the need to avoid sharing internal data with the external environment. To meet the company's needs, the proposed project considers the use of a private Blockchain, Hyperledger Fabric.
Despite its several advantages, Blockchain presents some limitations, for example,

it cannot handle big data and a huge volume of real-time information. Due to these limitations, the integration of the Blockchain in IoT applications is still at an early stage. For this reason, in this thesis, a first version of the project is proposed; thus, it will be possible to improve it by analyzing more types of data and by defining a more complex reward algorithm, for example by considering traffic analysis information in order to obtain more accurate results. To conclude, the choice of using the Blockchain with respect to other storage solutions leads to some benefits such as the warranty of data integrity and immutability. Furthermore, high levels of security and transparency are assured. To assure better decentralization, a hybrid blockchain solution could be considered: data are stored in the private platform and only the hashes of the blocks are stored in a public system. In this case, additional costs related to the insertion of transactions in a public platform have to be considered, too.

Acknowledgements

I would like to express my deepest gratitude to all those who have supported and contributed to the completion of this thesis.
I want to express a special thanks to my supervisor, Professor Danilo Bazzanella, for allowing me to carry out this thesis and for providing me with the necessary materials and resources.
Furthermore, I would like to express my gratitude to TurinTech for the opportunity and experience I have gained during the development of the thesis.
I am immensely grateful to my parents, who supported me during these years encouraging me to overcome challenges and reach my goals.
I would extend my thanks to my family for their continuous support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays a spread of new technologies is taking place. Among these technologies one of the most discussed is the Blockchain.
The objective of this thesis is to present how the Blockchain and Internet of Things (IoT) technologies can be implemented together. Specifically, this work has been carried out at TurinTech [1], an italian company which is investing in the Blockchain, in order to analyse some possible applications in Smart Cities and Automotive fields.
In Chapter 2, the IoT features, architectures and challenges are presented. A description of some types of IoT devices is proposed, especially of some devices which will be useful for the proposed application in Chapter 6, in order to define their features. Furthermore, some weaknesses of this technology are reported in order to better understand how the Blockchain can improve it. Then, some IoT applications in Smart Cities and Automotive domains are presented.
In Chapter 3, a brief excursus on the traditional databases solution is proposed. In particular, centralized, decentralized and distributed databases are presented and the CAP theorem is explained.
In Chapter 4, the Blockchain technology is introduced. After a brief presentation of its history and of its main elements, the basis of its functioning are reported. Specifically, the used hash functions and the cryptography are described in order to achieve a better comprehension of how Blockchain works. Thus, the different kinds of Blockchain platforms are reported, focusing on the mainly distinction between the permissionless and permissioned ones. Then, the main consensus protocols, such as Proof Of Work and Proof of Stake, are presented. A Section is dedicated to the Smart Contract, with a brief reference to ChainCode. In order to analyse how the Blockchain behaves with the external environment, the concept of oracle is explained, before theoretically, then with a short introduction to ChainLink. Thus, Bitcoin and Hyperledger are presented in order to provide an example of a public and a private platform, respectively. Specifically, one of the

main Hyperledger's project is presented, Hyperledger Fabric. Then, an example of possible architecture of a Blockchain application is proposed, specifically, a five-layer one. As for the IoT technology, advantages and weaknesses of the Blockchain are presented. Additionally, some of the most frequent malicious attacks are described, such as the 51% attack and the Sybil attack. Finally, a Section is dedicated to a comparison between Bockchain and traditional databases solutions.
Chapter 5 focuses on the integration of the IoT and Blockchain technologies. After a description of the role of the Blockchain in IoT systems, limitations and architecture of this integration are presented, in order to define the contexts in which is better to use these two technologies. Finally, some example of applications in Smart Cities and Automotive fields are provided, in such a way to outline some existing proposals.
In Chapter 6 a personal proposal of mine of application is presented. The aim is to provide a possible project involving the Blockchain and the IoT technologies for the Smart Cities domain. This thesis does not involve the implementation phase, but define the characteristics, tools and requirements to build the application. The choice of focusing on the documentation phase is because of the needs of the company which needs to evaluate the project, which has the purpose to highlight some advantages of the Blockchain, but that needs to be suited for a specific scenario in such a way to be more effective. Specifically, the project considers the public transport's vehicles and it aims to extract and store data in the Blockchain, in such a way to improve the quality of the public service and to minimize costs related to the vehicles, such as the fuel consumption. The proposed platform is Hyperledger Fabric, a permissioned one, chosen in order to meet the company's needs in terms of data privacy. The role of the IoT is to extract and analyse data from the vehicle before sending them to the Blockchain. Some examples of code are shown, in order to give an idea of a possible way to implement methods of the Smart Contract.

Chapter 2

Internet Of Things - IoT

2.1 Introduction to the Internet Of Things

Internet Of Things is one of the most popular technologies, which spread across several domains, plays a key role in the technological innovation.
After a brief description of the evolution of IoT (2.2), the two main architectures of IoT applications are presented in Section 2.3.
Because there exist several kinds of IoT devices, Section 2.4 proposed the description of three IoT devices: the sensors, the Smartphone and the Raspberry PI. Specifically these are three different types of IoT devices which cover a key role in a lot of applications.
In order to better understand this technology, its features and weaknesses are presented in Sections 2.5 and 2.6.
Finally, some examples of IoT applications in Smart Cities and Automotive domains are proposed (2.7).

2.2 History

Although Internet of Things (IoT) emerges in these last years, it is not a new technology. Indeed, in 1982 there was the development of one of the first IoT devices: a vending machine which, connected to the Internet, was able to inform users if it has got sodas before they go to buy one [2].
In 1999, Kevin Ashton defines the concept of IoT: "The Internet of Things connects everyday consumer objects and industrial equipment onto the network, enabling information gathering and management of these devices via software in order to increase efficiency, enable new services, or achieve other health, safety, or environmental benefits" [3].
From 1999, Internet of Thing has been an issue of continuous studies and researches.

In the last years, its diffusion has rapidly grown and has regarded several domains, as automotive, smart cities, industry and healthy. Specifically, some applications in the first two fields will be presented in Section 2.7.1 and 2.7.2.

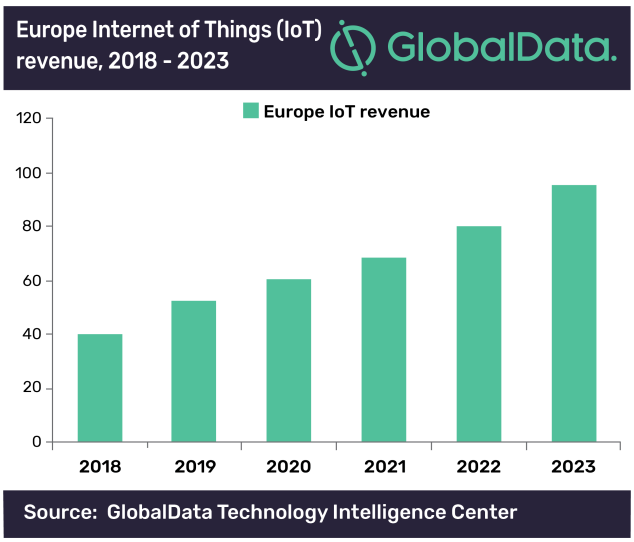

Figure 2.1: Europe IoT revenue, 2018-2023 [4].

2.3 Architecture

In order to better understand the applications of the IoT, the knowledge of its architecture is needed.
In [5] two different architectures are described: a three-layer solution and a five-layer one, presented in the two following subsections.

2.3.1 Three-layer architecture

In this architecture, the first level is the Physical one: it comprises the various type of IoT devices (such as sensors and smartphones). It aims to link and connect IoT devices in such a way to allow data transmission. There are different protocols and technologies used for this task, as RFID and NFC.
Thus, there is the Network layer. It is responsible for the transmission of data collected by IoT devices to the third level, the Application one. Different technologies can be used, as 4G,Bluetooth and WiFi.
Finally, the Application layer uses data in order to provide a service. It comprises API, user interfaces and functions to satisfy user requests.

2.3.2 Five-layer architecture

Because of the important development and evolution of this technology, the three-layer architecture proposed in the previous Subsection, it is not longer appropriate to manage the applications requirements. For this reason, a five-layer architecture is defined. The first level is the Sensing one and it comprises the IoT devices.
Thus, there is the Network layer, whose purpose is to allow the data transmission.
The Middleware layer aims to process and store data: because of the heterogeneity of IoT devices, it is necessary to guarantee compatibility among them.
The Application layer comprises a specific service for the user, while the Business layer manages the entire IoT system in such a way to control all the different services provided in the previous level. The fifth layer is also responsible for data analysis and creation of business models.

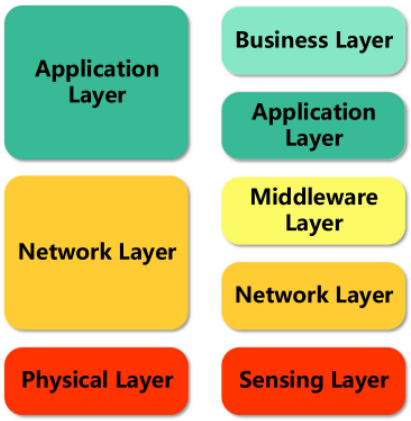

Figure 2.2: Three-layer and five-layer architectures [5].

2.4 IoT devices

In order to better understand the potential of this technology, a brief description of some IoT devices could be useful. Specifically, in 2.4.1 some type of sensors, while in 2.4.2 some features of a smartphone are reported.

2.4.1 Sensors

There are several types of IoT sensors and there are different ways to classify them. For example, they can be active or passive: if they need an external power source, they are passive, otherwise, they are considered active.
Sensors can be classified in analog and digital, according to the type of produced

signal.
Another possible classification takes into account the type of data obtained with sensors. If the presence of nearby objects needs to be monitored, proximity sensors are used. Temperature and humidity sensors measure the temperature and humidity of the environment. Acoustic sensors monitor data related to sound and vibration features. Electric and magnetic data are managed by electric and magnetic sensors. Furthermore, there are several other types of sensors, as pressure sensors, optical sensors and so on.

2.4.2 Smartphone

Nowadays, smartphones are considered to be an essential tool. In addition to the classic function of a phone, they can be used for several applications, thanks to its variety of IoT elements, for example, proximity sensors, microphone and camera. Through these elements, several functions such as the location monitoring, the facial and voice recognition, are possible.
Furthermore, smartphones can also be used as tool to transmit data, by supporting different technologies as WiFi, 4G and Bluetooth.
As it will be shown later, it covers an important role in a lot of IoT applications.

2.4.3 Raspberry PI

Raspberry PI [6] is a small single-board computer. It is developed by Raspberry PI Foundation and different versions are available.
Its several features, such as the small dimensions, the high flexibility and versatility and the powerful computer make Raspberry PI a device appreciated by specialists, but also by beginners.
Raspberry PI can support different connection protocols, such as Bluetooth, WiFi and Ethernet.
An essential feature is the presence of a General Purpose Input/Output (GPIO) which allow to connect and control a wide range of sensors.
Due to its characteristics, Raspberry PI is widely used in IoT systems, specifically, it is often chosen in order to manage all the nodes and sensors in a system. Therefore, it can be used to receive data from different IoT sensors, analyse them and the send them to a database or to the network. It is important to notice that it is compatible with a Blockchain system. Therefore it can play an interesting role in IoT-Blockchain applications.

Figure 2.3: Raspberry Pi 4 Model B [6].

2.5 Features of IoT

IoT devices present several features which differentiate them by the traditional devices.
Firstly, thanks to the connectivity of the IoT system, its devices can be accessible remotely and they can be connected each other, despite their diversity, thanks too to the supported heterogeneity, too. Indeed, an IoT infrastructure allow the connectivity and data exchange among different devices and differend protocols. These two properties, connectivity and heterogeneity, are essential because an IoT system involves a huge amount of devices, thus defining a large scale network. Because of the large size of the network, IoT devices have to be defined with a unique identified.
In order to support the heterogeneity, an IoT system is dynamic, and it collects a huge amount of data. For this reason IoT is often linked to the concept of Big Data that requires new types of methods and algorithms, in order to perform better analisys.
Furthermore, IoT devices present intelligent capabilities, through which they can make smart and autonomous decisions in different scenarios.

2.6 IoT Challenges

IoT systems present some challenges to be addressed.
One of the most critical is the security: a deeper study of the devices is essential in order to determine the vulnerabilities and adopt the proper security measures.

In case of a malicious activity, data collected by the sensors could be tampered. Furthermore, the problem of security is also linked to the privacy issue, because in most cases devices collect sensitive data.
If the IoT system is centralized, there is the problem of the Single Point of Failure: if the central server goes down, the entire system will go down.
Costs could represent a possible challenge: several types of cost have to be considered. For example, it is necessary to pay a trusted third party which manages the data storage. Furthermore, costs related to the maintenance of the hardware and to the software development are needed.

2.7 IoT applications

Nowadays IoT systems are applied to several fields.
For example, industries make use of IoT sensors for various purposes, such as the management of automated activities, the maintenance of security measures and the management of failures.
Another domain is Smart Home: there are a lot of IoT devices able to monitor some conditions of a house, as the temperatures, audio, lights, etc.
IoT devices can be used also in order to monitor some healthy data, like physical activity, sleeping hours and heartbeats.
In the following subsection, 2.7.1 and 2.7.2, some examples of IoT applications in Smart Cities and Automotive field are presented.

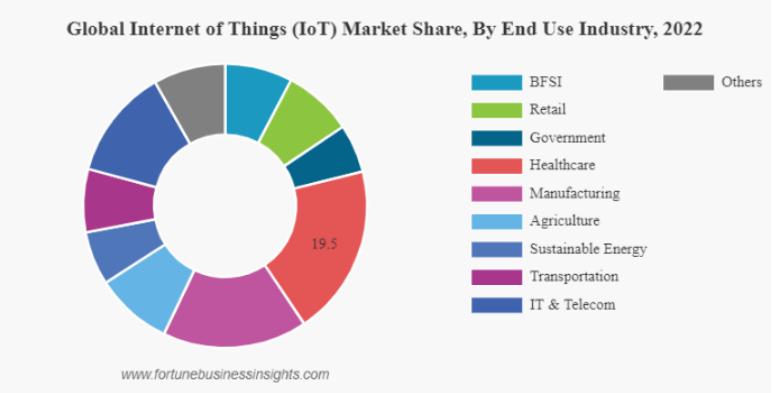

Figure 2.4: Main application domains of IoT, 2022 [7].

2.7.1 Smart cities

In [8] a solution to monitor parking availability using IoT sensors is proposed. In this case, an IoT sensor is placed in each parking place and monitors the presence

of a parked car. In this way, it is possible to show the real-time number of available parking places on a display.
In [9] a model to monitor the urban traffic behavior of electric vehicles is presented.
An urban environmental pollution monitoring is exposed in [10].
IoT sensors are also used to monitor the traffic in real-time in order to give information to users.

2.7.2 Automotive

IoT systems offer several services to the automotive field. IoT sensors can be positioned in a vehicle in order to offer additional services, such as real-time GPS services.
In [11], a model to monitor and detect collisions is proposed.
Another possible application is monitoring traffic in real-time in order to suggest the fastest possible path to the driver and eventually the presence of accidents.
IoT for the automotive domain has an essential role in the development of autonomous vehicles: IoT sensors and devices are put in the vehicle and, through machine learning and deep learning algorithms, they are able to recognize images and videos in order to make the vehicle more autonomous.

Chapter 3

Traditional Databases

3.1 A brief overview

Before introducing the Blockchain technology, a brief description of the three main types of databases is necessary, in order to better understand the novelty of the Blockchain and to make a comparison between the traditional storage and the new one.
Traditional databases can be centralized (3.2), decentralized (3.3) and distributed (3.4). After the descriptions of the these kinds of storage solutions, in Table 3.1, a brief comparison of the three types of databases is proposed.
Another important distinction is based on the structure of stored data: relational databases present a fixed schema, while non-relational (NoSQL) databases implement a flexible structure.
In a database different operations can be performed. Specifically, these operations are known as CRUD operations: Create, Read, Update and Delete. As it will be shown in Chapter 4, Blockchain does not support all CRUD operations and this aspect is one of the most distinctive of this new technology.
Finally, in 3.5 an important theorem is presented.

3.2 Centralized databases

In a centralized database, or central computer database system, data are stored in a single central server. The presence of a single location means that nodes of the network cannot directly communicate with each other; however communications and processes are mainly managed by the central server. Thus, the server is the central unit, while clients are the nodes in the network.
In order to access data, a Local Area Network (LAN) or a Wide Area Network (WAN) are needed.

Figure 3.1: Centralized storage.

The presence of a single central server allows to monitor and visualize data in a easier way. Furthermore data redundancy is minimal and data duplication is not needed. Due to the central management, the addition of new clients results easier. Because of the presence of a single central server, this system is relatively affordable.
On the other hand, the Single Point of Failure is a relevant weakness of this type of databases: in case of failure of the server, data are compromised or lost, thus resulting in the failure of the system. Furthermore, since the server is the only unit able to manage accessing operations, in case of high demand, the system might be overloaded, thus leading to slow responses.

3.3 Decentralized databases

In a decentralized network, the workload is distributed among some servers. Every server can manage the network in an indipendent way, as a central server. Thus, in this system there are several servers which offer services to clients.
This type of databases system allows to reduce the problem of the bottleneck, because in case of high demand, it can be divided among some servers. Furthermore, there is more tolerance to failures, leading to a higher availability in comparison to the one offered by centralized system.

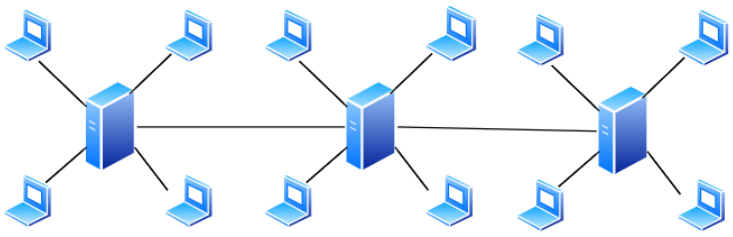

Figure 3.2: Decentralized storage.

A weakness of this network is the difficulty to effectively monitor the activities performed by each node: in some cases it could be difficult to understand which server has responded to a certain request. Furthermore, data might be lost during the transmission.

3.4 Distributed databases

In case of a distributed database, each node of the network contains data. This system is managed equally: nodes are generally peers-to-peers and the processing power is equally distributed.
The Single Point of Failure is solved in this system which generally also presents a higher security than a centralized network.
On the other hand, the management of a distributed database results more difficult.

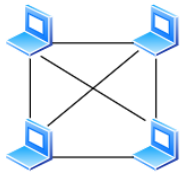

Figure 3.3: Distributed storage.

3.5 CAP Theorem

Distributed databases provide several advantages and features.
There are three main characteristics to be considered: Consistency, Availability and Partition tolerance, often indicated as CAP properties.
Consistency means that all nodes contain simultaneously the same information.
Availability allows to a client to always receive a valid response from the system.
Partition tolerance means that the system is able to work also in case of failures among nodes.
CAP properties are object of several studies, in order to better design distributed databases. Specifically, in 2000, professor Brewer presented the CAP Theorem, also known as Brewer's Theorem [12]. According to this theorem, distributed system can provide only two features among consistency, availability and partition tolerance.
This theorem acquires a relevant importance, especially for NoSQL databases. Specifically, they are classified based on the CAP properties guaranteed, into three

Table 3.1: Centralized vs Decentralized vs Distributed Databases

	Centralized	Decentralized	Distributed
Storage	Central server	Several servers	Peers-to-peers nodes
Data access management	Bottleneck risk	Load distributed among servers	Workload split up among all nodes
Single Point of Failure	Yes	No	No
Maintenance costs	Low	Medium	High

main categories: CP, AP and CA databases.
The validity of CAP theorem for the Blockchain technology is discussed in Chapter 4.

Chapter 4

Blockchain Technology

4.1 Introduction to Blockchain technology

Nowadays the Blockchain technology is the object of several researches and a lot of companies have started to implement it in their projects.
Blockchain is defined as a shared, immutable ledger. It can be useful in order to keep a history of transactions and to manage the tracking of assets which can be tangible or intangible.
In Section 4.2, a brief overview of the history of the Blockchain is presented, in order to better understand the evolution of this technology.
After a presentation (4.3) of the main elements of the Blockchain's structure, such as nodes and the wallet, Section 4.4 focuses on how it works, in a general way. The main phases of its work are explained, such as the validation of blocks and the nodes authentication.
Some security measures as the usage of hash functions and asymmetric cryptography are explained in Section 4.5. Specifically, the concept of Self Sovereign Identity (SSI) is introduced, in order to better understand how the Blockchain can manage the users' authentication without the need of a third party.
Although the Blockchain was born as a permissionless structure, over the years new kinds of platforms have been defined. In Section 4.6, the main types of Blockchain are presented.
As explained in 4.4, to validate a new block, nodes have to reach a consensus. The agreement can be obtained through a consensus protocol, which is the main focus of Section 4.7. In particular, the main consensus protocols are explained.
Thus, the concept of Smart Contract is presented in Section 4.8. Furthermore, a small description of ChainCode is reported.
Smart Contracts could require access to external information in a secured way. In order to meet this need, oracles can be implemented. A description of the features

of an oracle is proposed in Section 4.9 and a brief presentation of Chainlink is provided.
Bitcoin and Hyperledger are presented in Sections 4.10 and 4.11, respectively. In this way, examples of permissionless and permissioned Blockchain platforms are provided. Furthermore, the presentation of Hyperledger Fabric is included.
Thus, a five-layer architecture is proposed in Section 4.12.
The main advantages and disadvantages are exposed in Sections 4.13-4.14.
An overview of some Blockchain's vulnerabilities is presented in Section 4.15.
Finally, Section 4.16 focuses on the distinction between the Blockchain and the traditional databases solutions presented in Chapter 3, in order to better understand when it is convenient to use the Blockchain.

4.2 History

The concept of secured chain of blocks is presented by Stuart Haber *et al.* in 1991. In order to prevent tampering, they proposed a solution for the time-stamping problem in digitally timestamp electronic documents [13]. However, the Blockchain technology gains popularity at the end of 2008 and, in the following years, it devolops from Blockchain 1.0, Blockchain 2.0 and Blockchain 3.0.
Blockchain 1.0 is associated with the crypto currencies, in particular Bitcoin. The latter is defined at the beginning of 2009 by an anonymous person under the pseudonym of Satoshi Nakamoto [14].
Thus, Blockchain 1.0 is extendend into Blockchain 2.0. Specifically, it consists in the introduction of smart contracts, which are programs stored on a blockchain and their execution happens when some predetermined conditions are met. In Section 4.8 they will be described more in detail.
Thanks to the improvements of the Blockchain technology, it is possible to extend its applications beyond the monetary field, and it can be used for a lot of sectors and industries, as medical services, automotive industry and manufacturing. Thus, Blockchain 2.0 is extendend into Blockchain 3.0.

4.3 Elements of the Blockchain

Before explaining how the Blockchain works, a description of its main elements is needed.
This system consists of a set of nodes which are linked in order to define a decentralized network. Each node in the network contains a copy of the Blockchain. In a Blockchain network, a node can be a device, such as a computer, and it can be classified according to its functions or capabilities. The two most recurrent kinds are full and light nodes. The former store the entire chain and validate

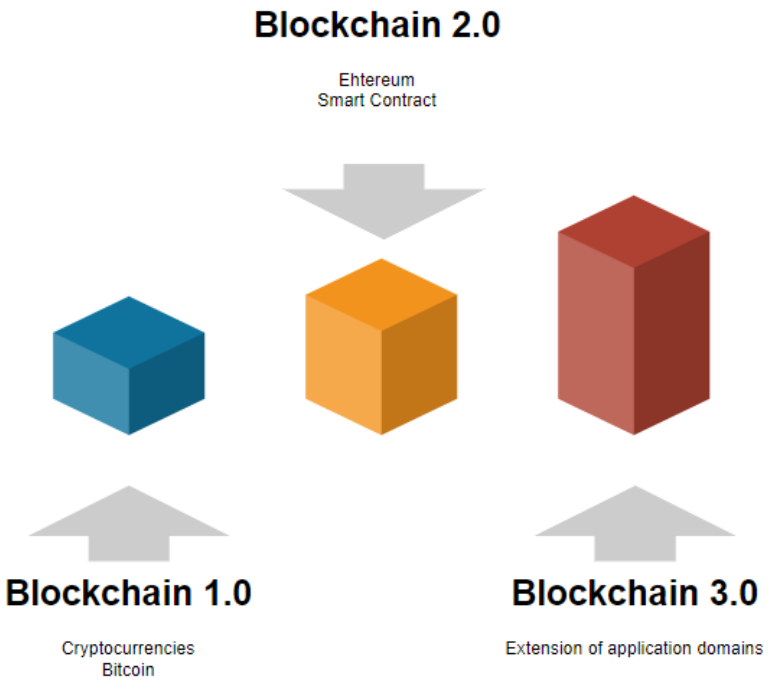

Figure 4.1: Evolution of the Blockchain.

transactions, while the light nodes only store the headers of the transactions and verify transactions only formally. A full node requires more computational and storage power.
Another important element is the wallet. It is a software, or hardware, which maintains a copy of the data related to the transactions done by the node. In particular, it contains the user's credentials and the public and private keys. There are various kinds of wallet, such as mobile wallets, desktop and web ones. For example, a mobile wallet is installed on the smartphone, while the desktop one is implemented on the computer. Furthermore, wallet can be custodian, if the security is supervised by the provider, or non custodian, if the security is a duty of the user.
As it will be shown later, in the Blockchain, transactions are inserted. Specifically, new verified transactions are inserted into blocks. A block is a structure in which all validated transactions are stored. Its structure can vary, but there are some elements that are common as the timestamp, the hash of the previous block and the Merkle root which is the hash based on all of the transactions in the block.

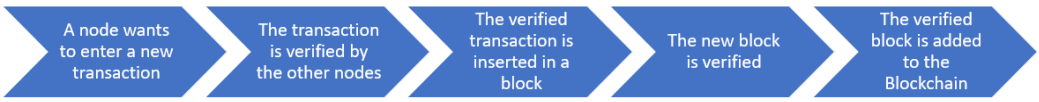

Figure 4.2: How Blockchain works.

4.4 How does Blockchain work?

When a node wants to enter a new transaction, it has to use the public and private keys to send it. Thus, the asymmetric cryptography is used, by means of private and public keys, in order to guarantee the security of the system.
Therefore, the transaction is verified by the other nodes and, if validated, stored in a block.
At this point the nodes of the network have to verify the block: specifically, it can be validated if it contains only verified transactions. When the nodes validate the new block, this is added to the chain, in which there are all the previous blocks.
In order to validate the block, an agreement among nodes is necessary. This agreement can be achieved through consensus mechanism, which will be described in Section 4.7.
Smart contracts can cover an essential role in the system, if supported by the used Blockchain platforms. For example, they can be implemented in order to automate some operations.

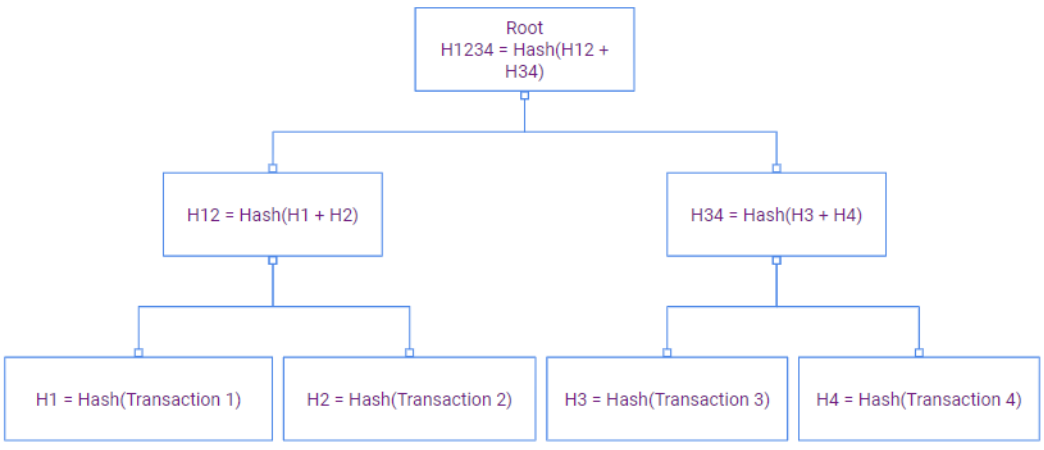

Figure 4.3: Merkle Root.

4.5 Hash Functions and Cryptography

As described in Section 4.4, Blockchain makes use of hash functions and cryptography. In order to better understand their role in the Blockchain technology, a brief description is reported in this Section.
A hash algorithm is a function which maps input data into a value, making original data unreadable. The hash function works in such a way that, in case of even a small change in data, the output value will result very different with respect to the original one. Another point of strength of this algorithm is that, given a hash value, it is not possible to obtain the original input data. Because of these features, hash functions are often used in order to guarantee data integrity.
The Blockchain technology adopts hash algorithms, in order to verify the integrity of transactions and blocks: as written in Section 4.4, each block in the chain contains the hash value of the previous block.
Blockchain makes use also of the cryptography, specifically, the asymmetric one. This type of encryption involves two different keys for each user, one private and one public. The public key is used to encrypt the message while the private one is used for the decryption. For the sake of simplicity, an example is reported: two users are considered, Alice and Bob. Alice wants to send a message to Bob and uses his public key to encode the message. Thus, when Bob receives the message, he uses his private key to decrypt it.
Therefore, in the Blockchain technology, the public key is published by users in order to receive payments and messages, while the private key is kept secret in order to verify the transactions.

4.5.1 The Self Sovereign Identity (SSI)

A concept related to the authentication of an user, is the Self Sovereign Identity (SSI) [15]. It is a digital identity that can be held by the user, without the need of a third party and it is based on the Blockchain. It involves three main elements: the Decentralized Identifier (DID), the Verifiable Credential (VC) and the DID document.
The DID is an alphanumeric string which is the unique identifier of an user. It is protected by the Blockchain's cryptographic system.
The VC is a document, such as the driver's license which can be stored in an immutable way, linked to the DID.
Finally, the DID document contains some additional information related to the user.

4.6 Types of Blockchain

There are different types of Blockchain. The main element used to classify this technology is related to the permissions. If everyone is able to access to the network, the blockchain is said permissionless. On the contrary, if the generation of a block requires to be authorized by some authority, the blockchain is said permissioned. In this case the access to the network is limited.
On the basis of this distinction, it is possible to define four main types of blockchain: public, private, hybrid and consortium.
In a public blockchain the network is decentralized and everybody can access to the system. Thus, it can be permissionless or permissioned. In the latter case, everybody can read the data but only a subset of users can insert new transactions. It is a Peer-To-Peer network, since each node has the same permissions of the others: it can generate and validate transactions and blocks. Bitcoin [14] is a public blockchain. Another famous example of public blockchain is Ethereum [16].
In the case of a private blockchain, the access to the network is allowed only to verified nodes. The network is managed by a central authority. It is permissioned. Another permissioned blockchain is the consortium blockchain. It is similar to the private one, with the difference that in this case the central authority is represented by a group. For this reason, there is more decentralization rather than in the private type in which the central authority is represented by one authority.
Finally, the hybrid blockchain presents both permissioned and permissionless mechanism. In this case, some actions will be open to everyone, while others will require the authentication of the user.

4.7 Consensus protocols

As introduced in Section 4.4, in order to validate a block and obtain a fault-tolerant system, nodes need to achieve an agreement. This can be reach by means of different mechanisms which are able also to guarantee the security of the system by validating and authenticating the transactions in the block.
The choice of the consensus protocol takes into account the type of application to develop. Specifically, there are four main categories: Byzantine Fault Tolerance Series, PoX Series, DAG Series and Ripple Series.

4.7.1 Byzantine Fault Tolerance Series

The Byzantine Fault Tolerance (BFT) consider a scenario in which Byzantine servers (nodes in the network) can arbitrarily modify the content of the transaction. If a system adopts this mechanism, it is fault-tolerant and it can work even if there are some malicious nodes or if some nodes fail.

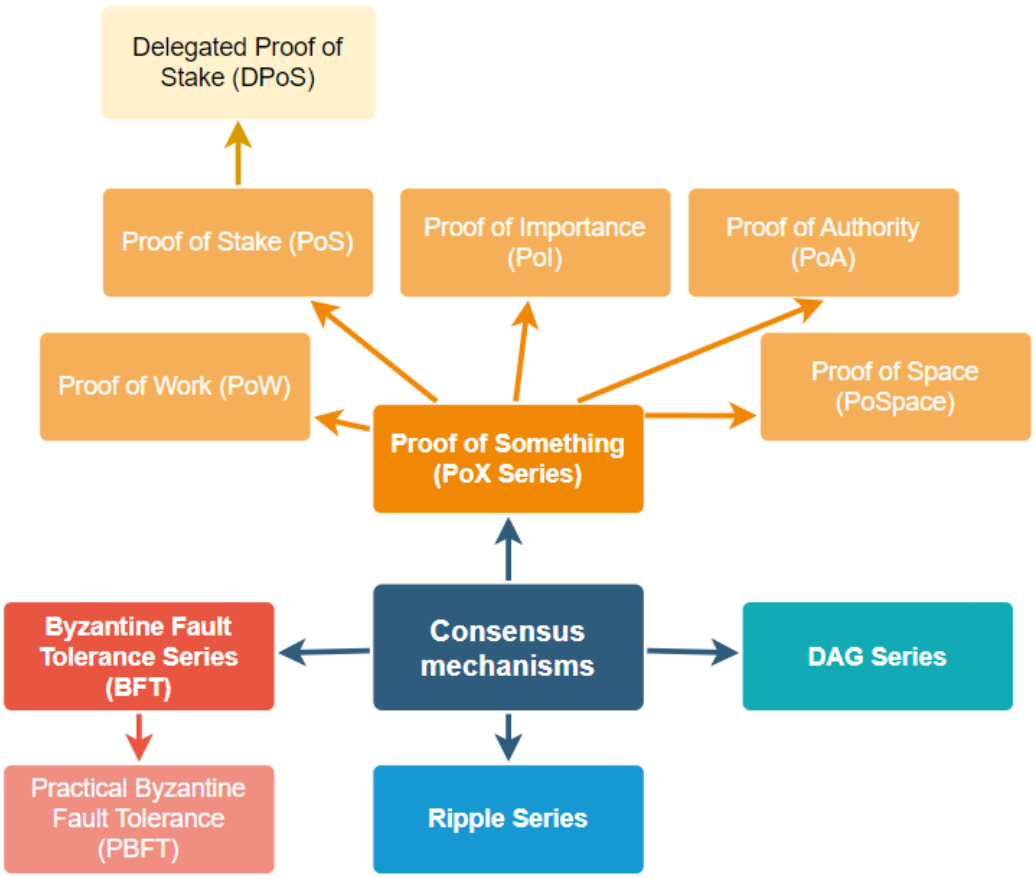

Figure 4.4: Taxonomy of the main consensus mechanisms.

One of the main solutions to this problem is the Practical Byzantine Fault (PBFT). Specifically, it is well suited for private blockchains. According to this mechanism, to validate a new block, every node of the network follows three steps, pre-prepare,prepare and commit. During each step, if the node is verified by at least $\frac{2}{3}$ of the other nodes, it will pass to the successive phase. This mechanism guarantees the security of the system. On the other hand, it presents some limitations in scaling to large networks.

4.7.2 Proof-Of-Something Series

Another category is the Proof-Of-Something (PoX) Series.
One of the main protocols of this category is the Proof of Work (PoW), presented

by Nakamoto in 2008 [14] and originally designed for Bitcoin. This mechanism allows to reach a consensus when all nodes solve a mathematical problem in order to validate the new block. Using this protocol, the chain of blocks results very resistant to tampering. This is because, in order to solve the problem, nodes need to have a large amount of computational power.
Another protocol of the PoX Series is the Proof of Stake (PoS). This mechanism is developed in order to provide an alternative to the PoW, which requires nodes with a large amount of computational power. The main idea behind this protocol is to take into account the amount of crypto-currencies owned by the blocks validators: nodes which possess large amount of crypto-currencies are more interested in avoiding malicious behaviours in order to guarantee the security. In this way, nodes owning more coins are more likely to be chosen to validate new blocks.
A variant of this mechanism is the Delegated Proof Of Stake (DPoS). In this case, a subset of nodes is chosen on the basis of the amount of coins of every node. This subset of nodes has the task to validate the new block. One of the main consequences of this protocol is the reduction of the decentralization of the system, because the nodes which are chosen to validate new blocks are a subset of the network.
Another mechanism is the Proof of Importance (PoI). It works in a similar way to the PoS, with the difference that, in order to validate and enter new blocks, also the exchanged coin is considered.
The Proof of Authority (PoA) takes into account the authority of nodes. Specifically, new blocks can be added only by nodes with authority. To do this, the knowledge of the identity of the nodes is necessary. Using this protocol the number of validators is small. This leads to a high scalability of the system. On the other hand, since the identity of the validators is known and the number of these nodes is small, it is easier to corrupt them and to try malicious attacks.
Another mechanism is the Proof of Space (PoSpace), also called Proof of Capacity. In this case, the validators are chosen according to the quantity of allocated memory on disk. The nodes with a large quantity of it are more likely to be chosen as validators.

4.7.3 DAG series

The third category of consensus protocols is the DAG Series. Thanks to implementation of DAG [17], there is an improvement in the scalability of the system and also in the transaction validation speed. This is because, in this case, every transaction is connected to the two previous transactions: thus, it can be validated via the previous two. It is important to notice that, because of the DAG structure, the chain is a tree: in this way a node can validate only the transaction with which

it is linked. DAG mechanism seems very efficient, altough it is still immature and it needs to be improved.

4.7.4 Ripple Series

Finally, the fourth category of consensus protocols is the Ripple Series [18]. In this mechanism, there is a distinction between Server and Client nodes. The former store the transactions and validate them. Specifically, a transaction is validated if more than 80% of agreeements are reached.

4.8 Smart contract

As written in 4.2, the Blockchain 1.0 technology evolves in Blockchain 2.0, thanks to the introduction of smart contracts. The idea of smart contract is initially presented as a combination of protocols used in order to secure the relationships among users [19]. With respect to a traditional system, the smart contract allows to secure relationships making use of codes and algorithms. It also allows to reinforce the overall security of a network, offering a protection against malicious attacks.
Thus, a smart contract be defined as an executable code and it aims to protect, guarantee and execute the terms of a contract without the need of a trusted third party. When predefined conditions are met, it runs on the blockchain network. Furthermore, a smart contract contains also a private database in which there are stored data. The platforms that support the implementation of the smart contract are several, as Hyperledger Fabric and Ethereum.
Smart contracts can also contribute to define decentralized authentication rules, in order to obtain a secured authentication system.
It is possible to divide smart contracts into two main categories, deterministic and non-deterministic. In the latter case, the contract needs to have external information, through services called oracles. An oracle is a service which looks for data to give to the system, after the validation. On the other hand, a deterministic smart contract uses internal data, thus it does not need external information.
Smart contracts are often used in IoT applications, as it will be shown in Chapter 5.

4.8.1 ChainCode

According to [20], ChainCode is "a program, written in Go, node.js, or Java that implements a prescribed interface". Thus, it can be described as a generic container of smart contracts.
It is used by Hyperledger Fabric [21] and it provides high flexibility. It has different purposes, for example it can define the data structures.

It is often used to define the business rules and functions that can be executed in the blockchain network.
Within the same ChainCode the implementation of several smart contracts is possible, and each smart contract is identified by means of an unique identifier.
In 4.1, an example of ChainCode written in Java is reported. Specifically, an example of a possible implementation of the definition of an Asset is reported.

Listing 4.1: ChainCode, Example of Java code in which the Asset class is defined

```
public final class Asset {
    //attributes of Asset class
    @Property()
    private final String assetID;

    @Property()
    private String ownerName;

    @Property()
    private String ownerSurname;

    @Property()
    private int postalCode;

    @Property()
    private int currentValue;

    //method to define a new Asset
    public Asset(final String assetID, final String ownerName,
                 final String ownerSurname, final int postalCode,
    final int currentValue) {

        this.assetID = assetID;
        this.ownerName = ownerName;
        this.ownerSurname = ownerSurname;
        this.postalCode = postalCode;
        this.currentValue = currentValue;
    }
    //add other methods, as get() and set() methods
    }
```

4.9 Oracle

In some cases a Smart Contract could need information from the external environment. Since a Smart Contract can read internal data, a sort of bridge with the external world is necessary. This bridge is called Oracle and it can be a software or hardware solution which certificates the trustworthy of external sources.

Specifically, the Oracle analyses external data, called off-chain data, and assures their correctness. Furthermore, it converts off-chain data into the proper format in such a way that they can be read by the Smart Contract.
For example, if the off-chain data to be analysed are the weather forecasts from the web, the Oracle will verify the web sites in order to validate and authenticate the information required by the Smart Contract.
There are several kinds of oracles. A first distinction can divide oracles into hardware and software. The former is a device, such as a sensor or a QR code reader, which converts data into a digital format and send it to the Smart Contract. An oracle can be implemented also as a software. In this case the Oracle is generally connected to Internet and it is used to check online data.
An important aspect to consider is the direction of the data flow: in some cases, off-chain data are incoming data, in other case they are outgoing and there could be scenarios in which there is a bidirectional data flow.
Furthermore, an oracle can be centralized or decentralized.
An oracle can be also a human, a qualified person who is responsible for off-chain data authentication.
Before implementing an oracle, a careful analysis of the scenario has to be conducted. In fact, the oracle's problem is an aspect that has to be taken into account. Specifically, since the oracle sends data to the Smart Contract, in the case in which the oracle is affected by a malicious attack, there could be security problems. Thus, in case of oracle's tampering, data could be corrupted. Another risk is related to the Man-in-the-Middle attack: a malicious attack might happen on the communication between the oracle and the Smart Contract.
Therefore, an oracle should have four features: trustworthiness, security, transparency and reliability. Thus, an oracle should provide untampered data, be secured against malicious attacks, allow users to verify how it validates data and use multiple data sources in order to obtain reliable data.
In the following Subsection, one of the most widely used oracle is presented, Chainlink [22].

4.9.1 Chainlink

Chainlink [22] is a decentralized oracle network. It is based on Ethereum platform and it validates off-chain data in such a way to guarantee trustworthiness. The data validation occurs with a decentralized approach, similar to the one used by Blockchain's validator nodes.
Chainlink makes use of a cryptocurrency, LINK, which is defined in order to compensate the nodes which validate off-chain data.
An important function of Chainlink is Chainlink Verifiable Random Function (VRF) [23]. Specifically, it is a verifiable random number generator (RNG): it randomly

generates some values together with a proof (similar to the PoX series used in Blockchain). In this way there is the warranty that data are untampered.
Chainlink can be used for several types of applications and it allows also the connection between the Blockchain and payment services.

Figure 4.5: Chainlink [22].

4.10 Bitcoin

Bitcoin [14], as written in Section 4.2, is created by Nakamoto at the end of 2008 and it is a digital, anonymous and distributed cryptocurrency. In this Section, a brief description of its characteristics is presented.
Its transactions are registered and validate through a public Blockchain. Since it is public, transactions are anonymized, in order to protect the privacy of the users.
The validations of blocks are conducted by miners and the adopted consensus mechanism is the Proof-Of-Work. Specifically, each miner has to create a new block with the verified transactions. Specifically, every transaction involves a fee, which is the amount that will be paid by the author of the transaction, in the case in which it is stored in the chain. Thus, miner has to iteratively define the hash value of the created block until the output value is less than a predefined target. The first miner able to obtain the proper hash value will insert the block in the chain. Furthermore, he, or she, will receive a reward and also the total amount of fees of the transactions into the verified block. If simultaneously more than one miner reaches a hash value lower than the target, a fork is created. Specifically, both miners will continue to operate on one of the two branches. Then, the longest branch will be probably chosen as the definitive one.
Bitcoin presents the limit of 1MB for the maximum size of a block. For this reason, miners decides which transactions to validate, often preferring those with higher fees. On average, the addition of a new block requires 10 minutes, in such a way to obtain the validation of a limited number of blocks, in order to avoid forks too long.
Furthermore, Bitcoin allows the implementation of smart contracts, explained in Section 4.8, but only in a limited way, for example it does not allow the definition of a loop.
Bitcoin presents some drawbacks, as the low scalability, the low validation speed and the high costs related to the Proof-Of-Work.

Figure 4.6: Bitcoin.

4.11 Hyperledger

Hyperledger is an open-source project proposed by Linux Foundation, introduced in [24].
It is modular, cryptocurrency-agnostic, interoperable, complete with APIs and highly secure.
Its modularity allows reuse of some components, by adopting extensible frameworks. In this way the changes of the system are possible without the need of completely redesign it.
Hyperledger presents a high level of security, thanks to a continuous update of the security adopted measures.
For what concerns the cryptocurrency, there is not the definition of a specific cryptocurrency, because, as underlined in the White Paper [24], Hyperledger's purpose is to "create blockchain software for enterprires, not to administer any cryptocurrency". Thus, the definition of a token is possible, but for the system functionality it is not compulsory.
Hyperledger supports the use of APIs, also among different systems. Furthermore it guarantees interoperability and portability, thus data exchange and transmission is possible also among different systems and Blockchain networks.
Thanks to these features, Hyperledger is well suited for a wide range of applications in different fields and domains, for example the industrial sector.
Hyperledger involves several projects such as Hyperledger Aries, Hyperledger Sawtooth, Hyperledger Indy, Hyperledger Fabric, etc.
Specifically, Hyperledger Fabric is presented in next subsection 4.11.1.

4.11.1 Hyperledger Fabric

Hyperledger Fabric [21] is a private and permissioned Blockchain: users have to authenticate themselves by means of the Membership Service Provider (MSP). As written before, it belongs to Hyperledger.
With respect to Bitcoin, it allows a very flexible implementation of Smart Contracts.

Specifically, they are defined by using ChainCode [20]. For what concerns the consensus protocols, Hyperledger Fabric provides a wide choice: users can choose the mechanism they prefer according to their needs.
Hyperledger Fabric presents a good level of scalability, thanks to its characteristics as the modularity and flexibility. For this reason it can be chosen by companies which want to adopt the Blockchain technology while preserving the privacy by means of authentication and permissioned networks.
As written before, in order to verify the identity of nodes, the Membership Service Provider (MSP) is used as a validator system. There exist the local MSP and the channel MSP. The former operates at the node level while the latter involves the channel level. Thus it requires the list of all the MSP of the entities belonging to the channel.
Another important aspect of Fabric, related to the MSP, is the Certificate Authorities (CA). CA, belonging to X.509 certificates, are defined in order to allow to organizations to access to the network and the MSP uses them in order to verify the identity of an entity.
Due to its permissioned nature, Hyperledger Fabric allows the implementation of hierarchical roles: in order to cover the companies' requirements, it is possible to define which resources each node can access.
Fabrics also allows creating channels. This feature is important because there are cases in which users prefer keeping private their transactions, for example because they are competitors; the creation of a channel aims to protect these transactions, in such a way that participants can access only to information stored in their channel, while keeping private data exchanged in other channels.
Another essential aspect of Fabric is the endorsement policy, which defines the set of nodes that are responsible for executing smart contracts (ChainCode [20], 4.8.1), in such a way to define which entities are essential to validate transactions. Through this policy, if a transaction needs to be validated by a specific node, in the case in which this node does not verified it, this transaction will be labeled as invalid.
The endorsement policy covers a key role also in the consensus mechanism. Specifically, Fabric introduces a new paradigm, execute-order-validate, while most of the other Blockchain platforms support an order-execute one. In the execute phase, a transaction is executed and verified. If the transaction is correctly validated, a consensus mechanism orders the transactions, which are then validated according to the endorsement policy. Therefore, validated transactions are added to the chain.
Belonging to Hyperledger, Fabric does not require the definition of tokens or cryptocurrencies, that eventually, if useful for the application, can be integrated into the system.
Thanks to all these features, Fabric is well suited for a lot of industry applications, and for this reason it is often chosen for different types of projects.

Figure 4.7: Hyperledger Fabric.

4.12 Blockchain architecture

In order to better understand the Blockchain implementations in IoT applications, a general description of the architecture is needed. The number of layers can vary, depending on the type of considered application. In [5], a five-layer architecture is proposed.
The first layer is the Physical one. It is the set of nodes, that linked together, define the Blockchain network.
Thus, there is the Network layer which allows the communication between the nodes. Its aim is to guarantee the connectivity of the network in such a way that all nodes are updated and synchronized.
Since in most of the cases, as already written, the Blockchain network is decentralized and there is not a central server, a Consensus layer is necessary: in this way, nodes can operate and take decisions reaching an agreement by means of consensus mechanisms.
The fourth layer is the Propagation level, in which protocols to establish how to propagate and store blocks are defined.
Finally, in the Application layer the functions compliant with the application requirements are defined. This layer can be often divided into two sub-levels: the presentation layer and the execution one. In the former, APIs, user interface and scripts are implemented, while in the execution layer transactions are executed and smart contracts can be included.
It is important to underline the fact that, the architecture just described is not fixed, but it can change according to the needs and type of applications.

4.13 Advantages of Blockchain

The Blockchain technology presents several benefits. Many of them depend on the kind of the implemented application, while others are presented in most of the cases. This section describes some of the advantages that are rather common in different types of applications.

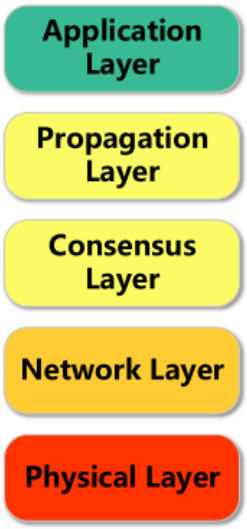

Figure 4.8: Five-layer architecture [5].

Firstly, the Blockchain technology guarantees the immutability of the stored data. Transactions can not be modified unless the majority of the nodes reaches an agreement. Furthermore, if a malicious node tries to change the message of a transaction, it will fail because the hash function stored in the block will be very different with respect to the hash code of the original block that is stored in the other nodes; the only possibility should be changing the content of every node but this is quite impossible thanks to the structure of the network. For this reason, the Blockchain is tamper-proof and, since each node stores a copy of the chain, in the case of the failure of some nodes, data loss does not occur (failure resistance). The high level of security is also due to the use of the asymmetric cryptography.
Since each block contains a timestamp, there is an improvement in the traceability and transparency of stored data.
Another advantage consists into removing the need of third parties: by using the consensus mechanisms the network is able to reach an agreement without the need of a central server. Additionally, the Blockchain network has a decentralized structure in which the propagation of information is managed by means of different protocols, as the Gossip protocol, and every node stores a copy of the data: in this way the need of third parties is useless. Furthermore, the necessity of a third party is often due to the lack of trust among users: it is possible to replace the third party with the Blockchain technology thanks to its high level of security, as written before.

4.14 Disadvantages of Blockchain

In Section 4.13 some of the main advantages in adopting the Blockchain system are presented. On the other hand, this technology has some disadvantages.
Firstly, due to the decentralized structure and to the consensus mechanism, a high level of energy consumption occurs. For example, the PoW needs a high power consumption: for this reason the choice of the proper consensus protocol is very important, in order to reduce the computational costs. Furthermore, the high level of energy consumption is due also to the fact that every node stores a copy of the chain. Because of this, there is the necessity of having nodes with a sufficient capacity in order to collect all the data. This requirement leads to some scalability issues, especially when facing with applications that stores a lot of data as for some IoT services, as it will be shown in Section 5.3.
Another problem arises when it is necessary to store a lot of data in real-time applications: in order to store a transaction, nodes have to reach an agreement through a consensus mechanism. This phase requires time: for instance, by using Bitcoin is possible to create a new block in 10 minutes approximately. Because of this, the processing speed is not high and this could lead to some problems.
Other disadvantages are related to the lack of standards and the difficulty to regulate the Blockchain systems with legacy systems. These two drawbacks are due to the decentralized structure and to the novelty of this technology.

4.15 Vulnerabilities of Blockchain

Despite the high level of security, the Blockchain technology presents some vulnerabilities to be taken into account. In this Section some of these vulnerabilities are presented, in order to have a better idea of the technology.
The 51% attack is one of the most critical attacks that affect Blockchain. It can happen when there is a group of miners who own more than 50% of the network's mining hashrate, i.e. the computational power. In this case, this group could manage the entire network because it can dominate decisions regarding blocks validations. Furthermore, they can also reverse transactions in order to double-spend coins. This attack can be prevented by avoiding the presence of a miner who owns a large portion of the computational power.
Another important attack is the Sybil attack: in this case, a single node creates fake accounts in order to obtain more computational power and have an high level of influence in the network. One of the consequences of this attack is the 51% attack: if the malicious node is able to obtain more than 50% of the computational power, this attack leads to a 51% one. Another consequence is the possibility of the malicious actor to manage the transactions deciding when validating them or

not. It is possible to prevent this attack by performing identity validation, in order to correctly identify nodes in the network.
A third example of malicious attack, is the Denial Of Service Attack (DoS). It is generally less critical than the Sybil attack, because its effect is temporary. Specifically, it temporarily causes the inefficiency of the Blockchain network, by sending a huge number of requests to the system, in order to overload it. A possible way to avoid it is to add new nodes across several places, in order to restrict the size of the memory queue.
Another critical attack is the Eclipse attack. In this case, a malicious person chooses to isolate a specific node by creating a fake environment, which often results in a peer-to-peer network. Thus, the malicious actor can control and manipulate the target node in such a way to validate transactions which otherwise would not be validated. A possible measure that could be taken in order to prevent this attack consists on restricting the number of connections of a node, specifically the inbound ones.

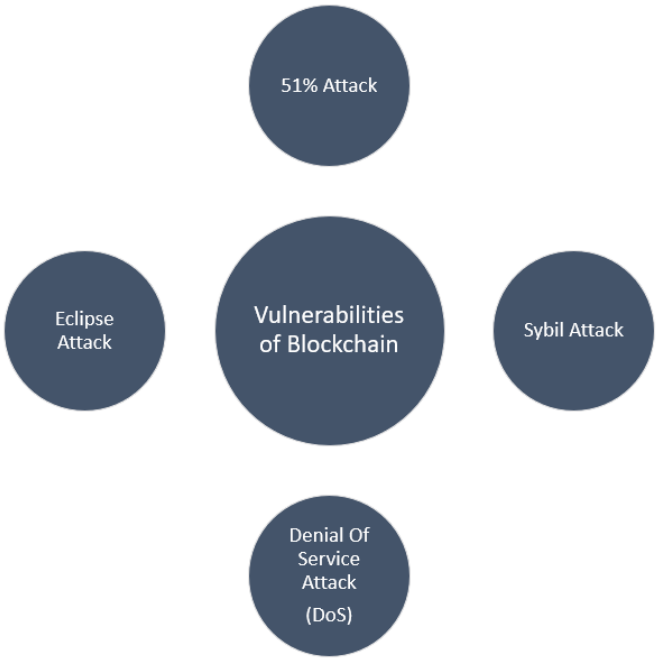

Figure 4.9: Some of the main vulnerabilities of Blockchain.

4.16 Blockchain or traditional database?

As reported in Section 4.13 and 4.14, Blockchain technology presents both benefits and drawbacks. Therefore, there are cases in which it is convenient to adopt it while in other applications the adoption of one of the three traditional databases, presented in 3.1, is more appropriate.
When there is the need to keep historical data, Blockchain results better because it allows to store all data. Furthermore, it protects data against tampering: since each node contains all the validated blocks, it is unlikely to change the content of all nodes in the chain, moreover, as said before, stored data in the Blockchain are immutable, thus only creation e read operations are supported.
Another scenario in which Blockchain leads to several advantages is when there are multiple and different parties that communicate each other: as written in Section 4.13, with respect to a traditional database, the presence of a trusted third party is not more useful.
Blockchain can be an optimal choice also when it is necessary to certificate the quality of a product, for example to demonstrate that it is complying with a standard.
An example of a scenario in which the choice of a traditional database should be better, is the case in which there is only one involved party. Furthermore, if there is not the need of storing historical data maybe a traditional solution could be better.
In order to choose the right database implementation, it is also important to consider which of the three CAP properties are necessary for the application. The Blockchain technology supports two of these properties, availability and partition tolerance. On the contrary, consistency is not supported, even if it is possible to say that evenutal consistency is covered by Blockchain, depending also on the type of the adopted consensus mechanism.
To summarize, the choice of implementing the Blockchain technology or a traditional database depends on the requirements and type of application considered. In Table 4.1, a brief comparison between Centralized Databases and Blockchain is reported.
A similar analysis has to be done in order to establish the type of Blockchain to adopt. Firstly, it is necessary to decide whether to choose a permissioned or permissionless system. Thus, it is important also to establish if to use a public or private solution.
After the proper analyses, the choice of the platform is fundamental. These evaluations should be done by considering the technical requirements and the purpose of the application.
In Table 4.2 a comparison between Ethereum and Hyperledger Fabric is reported.

Table 4.1: Centralized Database vs Blockchain

	Centralized Database	Blockchain
Authority	Central authority	No need of a trusted third party
Storage	Central Server	Every node stores all data
CRUD operations	All	Update and Delete are NOT supported
Timestamp	Optional	Mandatory, automatically added
Integrity	Malicious attacks can affect data integrity	Supported
Performance	Faster	Slower

Table 4.2: Ethereum vs Hyperledger Fabric

	Ethereum	Hyperledger Fabric
Type	Public	Private
Permissions	Permissionless	Permissioned
Currency	Ether	None
Consensus	PoW or PoS	Different mechanisms can be adopted
Smart contract	Solidity, Vyper...	Go,JavaScript...
Transactions per second	20	20000

Chapter 5

Blockchain integrated in IoT systems

5.1 A brief overview

In this Chapter a possible role of the Blockchain in IoT projects is analysed.
In fact, Blockchain could contribute to mitigate some of the weaknesses of IoT systems, as shown in Section 5.2.
Despite the advantages of this integration, there are some limitations to take into account (5.3), such as the scalability.
A five-layer architecture of a Blockchain-IoT application is presented in Section 5.4.
Finally, some examples of projects involving the Blockchain applied to IoT systems are reported in Section 5.5.

5.2 Blockchain's role in improvement of IoT systems

Nowadays IoT devices are expanding into several sectors.
As discussed in Chapter 2, IoT systems lead to several advantages, but there are also some vulnerabilities and weaknesses which could cause problems of different nature. In Table 5.1 some of the challenges of IoT systems, already described in Section 2.6, are reported.
In order to solve these problems, different studies and researches are conducted. One of the considered technology is the Blockchain, whose features are useful to address some of the IoT weaknesses. Specifically, the integration of these two technology allows to benefit of the advantages of Blockchain and go beyond the limits of the IoT systems.

Table 5.1: Some challenges of IoT technology

Feature	Description
Security	Security measures are necessary in order to protect data integrity
Privacy	In a lot of cases, stored data contain sensitive information
Single Point Of Failure	In case of failure, the system may be down
Trusted Third Party	Necessary to guarantee trust in the system

Due to the taken security measures, as the hash algorithm and the asymmetric cryptography (4.5), Blockchain can guarantee a better protection against malicious attacks. Furthermore, it allows to keep historical data and it protects them against tampering. In this way, data integrity can be warranted while IoT systems alone are not always able to guarantee it with the same strength of the Blockchain. Since IoT devices often collects sensitive information, data protection is essential in order to maintain privacy of users. In order to do so, anonymization is important: by encrypting messages, Blockchain can assure a good level of data security. For what concerns data privacy, it can be enhanced in a stronger way by adopting a permissioned platform.
Furthermore, the Blockchain technology ensures an efficient identity management: in order to access data and send transactions users have to register by using their credentials. In the case of a permissioned Blockchain platform, these controls aim also to limit the access to the network only to users with the right permissions. IoT devices access can be monitored by means of smart contracts whose methods can be defined in order to verify identities, also in a decentralized way.
For what concerns the Single Point of Failure issue, since IoT systems are often based on a centralized server: in case of failure, the entire system will be down. By using the Blockchain technology, the system will be based on a decentralized and distributed network which will guarantee the proper functioning of the system even in case of one or more point of failure.
Blockchain also allows to reduce the costs related to the maintenance of a Trusted Third Party (TTP): because of its decentralized structure and the transparency of the system, Blockchain does not require a TTP. In this way data transmission and storage are more efficient and secure.
The benefits due to the integration of Blockchain with IoT systems are several and in this Section only the most important are described. It is essential to notice that

other advantages are provided but they depend on the type of application and on the considered domain.

5.3 Limitations

Despite the several advantages, the integration of Blockchain with IoT technology presents some challenges that have to be taken into account before the development of an application.
Firstly, a critical aspect is to define a right balance between transparency and data privacy. One of the most important aspects of the Blockchain technology is that Blockchain can guarantee transparency of transactions, but this can affect the users privacy: in a public Blockchain, all transactions are visible and this could lead to some privacy issues. In order to protect data from external entities, a lot of applications are based on private and permissioned Blockchain. For example, Hyperledger Fabric allow creation of channels in order to better protect information. Before developing an application, a careful evaluation of right equilibrium between transparency and data privacy is essential.
While IoT devices present a limited bandwidth and resources, Blockchain requires high power consumption: this could lead to performance problems, since nodes of the Blockchain require a lot of resources to store data and validate blocks, in order to guarantee security. Because of this, the definition of a good trade-off between performance, power consumption and security is essential.
Another challenge linked to the limited bandwidth and resources of IoT devices, is related to the Big Data field. Unfortunately nowadays Blockchain does not seem to

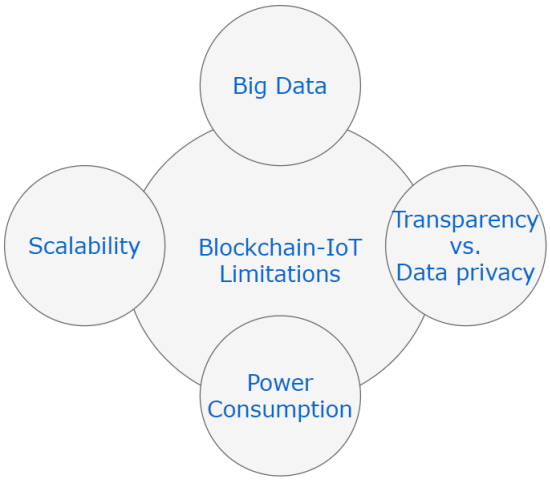

Figure 5.1: Some of the Blockchain-IoT limitations.

be good at handling Big Data. This is due to different reasons. Firstly, since each node of the Blockchain stores all data, it has to be equipped with a large capacity resources. In addition to this, the limited throughput of the Blockchain, due to the consensus mechanism and the adopted cryptography, acts as a bottleneck. Therefore real-time storage of a huge volume of data is not possible. This limitation is very critical because in a lot of cases IoT sensors send a huge volume of real-time data to the network and the Blockchain technology can not handle Big Data.
Another challenge is related to the scalability. IoT systems have high scalability while the Blockchain presents a limited scalability and it does not work efficiently in case of large-scale network.

5.4 Architecture

The architecture of an application in which Blockchain and IoT are integrated, can vary and different solutions are proposed. In this Section the solution proposed by [5] is presented.

5.4.1 Five-layer architecture

The described architecture involves five layers.
The first level is the Physical one and it comprises IoT sensors and IoT devices, which are data generators. This layer is similar to the physical level of the IoT architecture presented in 2.3.2.
Thus, the Network layer manages the communication among IoT devices, specifically it involves the routing, multicasting and internet working operations.
The third level is the Blockchain layer. It consists of the Blockchain platform, thus it includes consensus mechanism, data sharing and data storage. In particular, Blockchain can be used ad a distributed database.
Then, the middleware layer is the fourth one. It manages the Blockchain network and provides additional security measures. Therefore, its main aim is not the integration of IoT devices, but its purpose is to focus on the Blockchain management.
Finally, there is the Application layer: it is responsible for interactions between the system and the user. It involves also APIs and abstraction services.

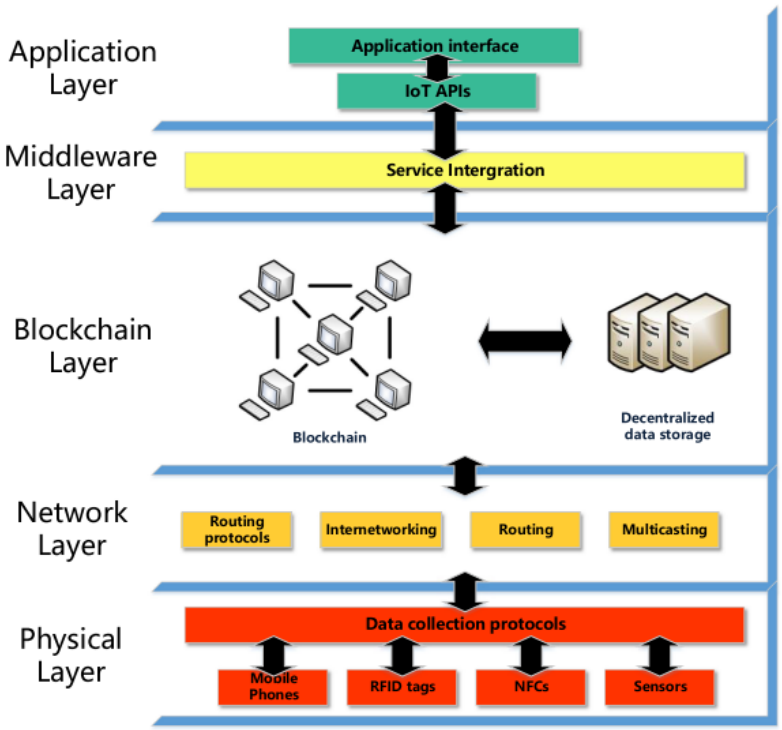

Figure 5.2: Five-layer architecture [5].

5.5 Applications

5.5.1 Smart Cities

Tokyo, Daimaruyu district: the first smart city based on Blockchain

In 2018 the first Blockchain based smart city was published. The experiment involves a district of Tokyo, Daimaruyu, composed of three quarters for a total of 120 acres [25].
The principal aim of the experiment is to build a Blockchain network in order to manage the several data produced by the environment of the smart city and convince companies about the importance and efficiency of the data storage and data security based on a Blockchain system.
The main responsible of the project, Fujitsu, chose Hyperledger Fabric as Blockchain platform. He defined Virtuora DX [26], in order to allow data sharing and smart contracts among participants. By using a private platform as Hyperledger Fabric, it is possible to isolate data within each company, thus protecting data privacy.

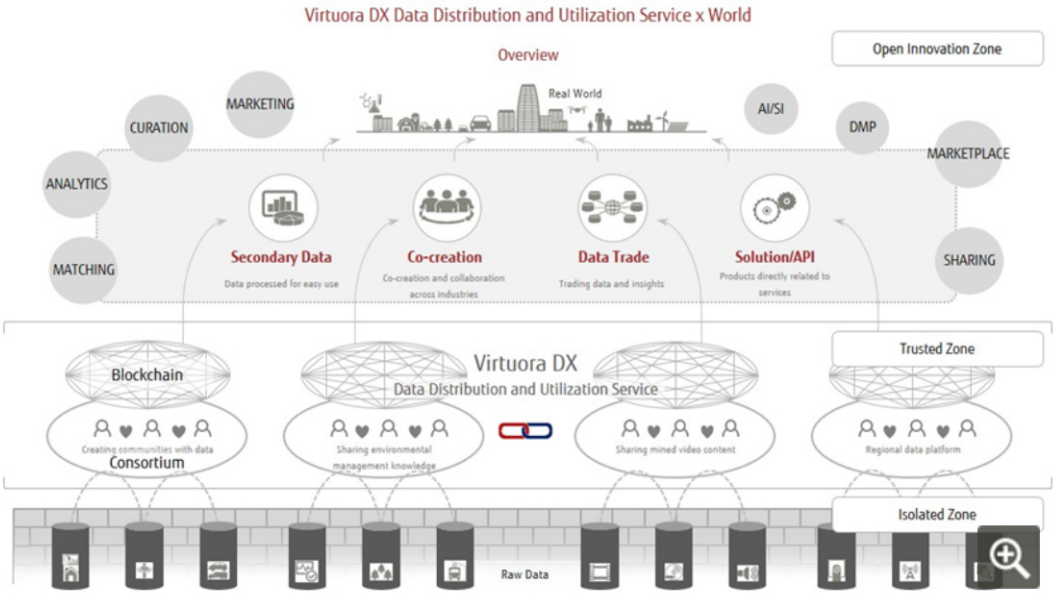

Figure 5.3: Fujitsu's proposed Data Distribution and Utilization overview, [26].

Parkchain

An example of possible application in which IoT and Blockchain technologies are integrated togheter is Parkchain, presented in [27]. This project is defined with the purpose to propose a possible solution to mitigate the problem of finding parking: nowadays, due to the growth of number of vehicles, finding parking spaces becomes more and more difficult, thus leading to other issues, such as the increasing of polluting emission and traffic congestion.
In paper [27] authors propose to implement the IoT and Blockchain technologies in order to allow private people to rent out their unusued grounds as parking places. IoT devices aim to get and analyze images sent by camera surveillance units, which monitor the area. Specifically, IoT devices examine images by means of a software based on Python libraries, in order to define the presence or absence of a vehicle in the parking place. Results of these analyses are stored in a relational database.
Users can find an accessible parking place by means of an application, installed on their smartphone. Through this application, users can also reserve a parking place and proceed to pay for the service.
Data related to the parking activities are stored in the Blockchain system. Specifically, smart contracts are used to manage transactions. Furthermore, in order to connect Blockchain with the database, an oracle is implemented. The chosen Blockchain platform is Ethereum.

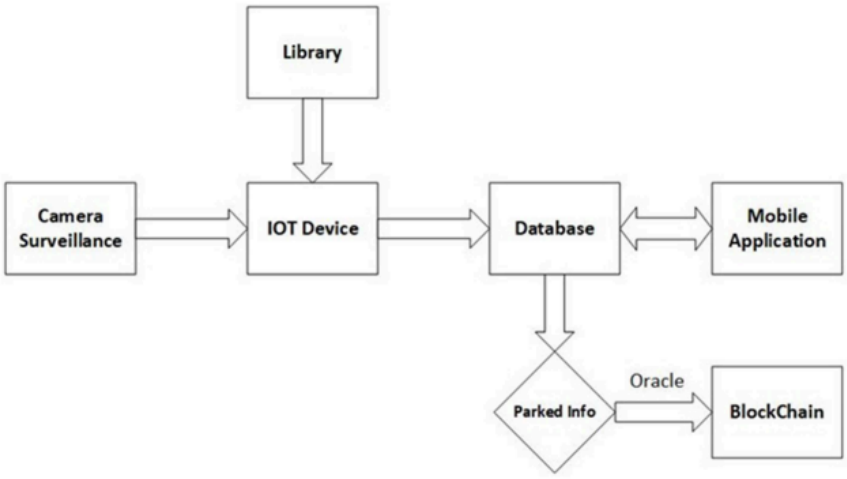

Figure 5.4: Proposed surveillance system proposed in [27].

Waste Management

Waste Management involves all the different steps related to the waste, such as the collection and transportation phases, but also the legal and economic aspects, as the regulation of the process.
Nowadays Waste Managament covers a critical role in our lives and leads to several consequences in different fields, like the environment, the health and quality of life. For this reason, a lot of studies are conducted in order to improve the Waste Management and make it more sustainable.
Some of these researches propose Blockchain based solutions, as Swachhcoin Foundation [28]. In the White Paper [29], the project is explained.
Its purpose is to obtain an economic value by transforming wastes in a sustainable and eco-friendly way. In order to perform this objective, Blockchain and IoT technologies are integrated together.
IoT is implemented for the remote control of machinery and also for remote assistance to the controller.
Data are sent to Ethereum, the chosen Blockchain platform. Furthermore, some Smart Contracts are implemented in order to manage the network according to predefined rules defined in a transparent way.

5.5.2 Automotive

An electric car based on Blockchain

IndiEV [30]-[31] developed Indi One, an electric car based on the Blockchain technology. In [32]-[31] the manner with which Blockchain is implemented is briefly described. Specifically, each Indi One car represents a node in the Blockhain network, in such a way that, for every new sold vehicle, a new node is added to the chain, thus leading to a large network. In order to obtain such scenario, in each car, a Vehicle Integrated Computer is implemented. In this way it is possible to connect the physical world with the digital one.
The objective of this project is to allow a personalized experience of the owned car. Therefore owners can benefit of a passive income generated during the use of their car.
Furthermore, IndiEV developed also a Blockchain based APP store through which vehicle's owners can make the transactions.

Electric vehicle charge

Another interesting project involves the charge of electric vehicles. By using the Blockchain technology and IoT devices an electric vehicle is able to connect with the charging column. This project is proposed by BMW [33] in [34].
The principal aim is to facilitate and speed up the identification phase: by means of this system, drivers do not have to identify themselves through a smart card, but the identification phase happens in an automatic way thanks to the use of smart contracts.
BMW also proposes to use a similar system for autonomous vehicles in order to make them independent from people for the recharge.

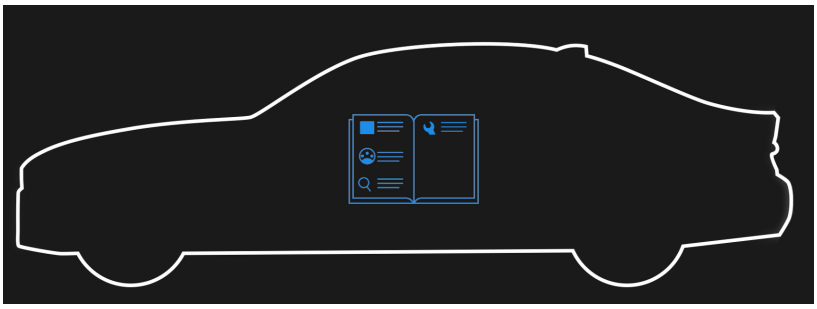

Figure 5.5: [34]

Chapter 6

Case study: A possible project involving IoT and Blockchain to optimize Public Transport

6.1 Introduction

In the previous chapters IoT and Blockchain technologies are introduced. After their presentations, the benefits and weaknesses of their integration are described. In this chapter, a possible application is presented. Specifically, it is a my personal proposal and it involves the Blockchain and IoT applied to the Smart Cities domain, in particular for the public transport.
The architecture of the project is the five-layer one presented in 2.3.2.
In the following Sections, the proposed project is described. After the presentation of the main objectives and scenario, the IoT system and the Blockchain platform are shown. Thus, the several functionalities are presented, such as the Reward system.

6.2 The project

6.2.1 As-Is scenario

Nowadays the public transport plays a critical role in the urban environment. In the last years governments often offered incentives to the citizens in order to increase the usage of public transport and, respectively, to reduce the use of private vehicles,

thus aiming to obtain some advantages, such as the decrease of traffic congestion and of the pollution.
For this reason the continuous improvement of the service is essential, in order to convince people to adopt the public transport solution. Unfortunately, there are some challenges, such as the difficulty to guarantee the timeline and the high costs related to the maintenance of the vehicles.
As it will be shown later, one of the purposes of this project is related to the effort of decreasing the costs related to the maintenance of the public vehicles.

6.2.2 To-Be scenario

Objective and possible advantages

The main goal of this project is to extract data from the vehicle and store them in the Blockchain network. Data are extracted in order to analyse and monitor the driving style and encourage the driver to drive better by means of some incentives.
The choice of improving the service by monitoring the driver's driving style is made because it leads to several advantages.
Firstly, if the driver drives well, it is likely that maintenance costs related to the usage of the vehicle and to the fuel consumption go down. Furthermore, in this way the maintenance services can be saved and monitored.
Then, people are more likely to take a bus if they perceive that the the driving is quite safe, while they could choose not to take it if they notice a poor driving style.
Another possible advantage is that it is possible to monitor the level of pollution produced by the vehicles, such as the CO_2 emissions.
In order to encourage drivers to drive better, it is possible to use some incentives, in such a way that also the driver can better recognize the advantages of the project.
As it will be shown later, through Smart Contracts it is possible to give prizes and benefits proportionally to the quality of driving. Some of these benefits could be, ticket restaurant, vouchers related to the purchase of books and movies and wellness gift packages.

A road map

The purpose of this thesis, as already written, is to provide a presentation of the Blockchain technology and describe how it is possible to apply it on IoT systems in the Smart Cities domain and a possible application to implement. Thus, the proposed project aims to provide some technical guidelines and suggestions. In this thesis a first version is presented, but it could be integrated with more complex algorithms, functions and tools in order to improve it. The choice of presenting a first "smaller" version is due to the current requirements: the main purpose is

to study its feasibility and a simpler scenario has been preferred. This version tries also to provide a possible way to overcome one of the limitations presented in 5.3, specifically, how to manage a huge volume of data without overloading the Blockchain network. Thus, it will be possible to improve the project by integrating new functionalities. At the end, some possible improvements will be presented.

Data classes

Before starting describing how the project works, a description of the defined data classes is necessary.
There are three classes, Vehicle, Driver and Service.
Vehicle class is composed of different attributes such as VehicleID and MotorType, which represents the type of motor (electric, fuel...) and the SSI . The key is VehicleID which corresponds to the identification of the bus (or tram).

Driver class has as key the DriverID which corresponds to an unique code

Table 6.1: Vehicle class' attributes

Attribute	Type
VehicleID	String
MotorType	String
SSI	String

assigned to each dependent (the SSI). Other attributes are: Name, Surname, Password,Email.

Finally, the Service class represents the specific service and it presents several at-

Table 6.2: Driver class' attributes

Attribute	Type
DriverID	String
Password	String
Name	String
Surname	String
Email	String

tributes, such as the StartService and EndService, the DriverID and the VehicleID, Oxygen, MeanVelocity, NumberChanges, MaxPeak, EngineRPM, MeanFuelConsumption, BatteryStatus and DriveLevel. Specifically, DriveLevel is an integer which corresponds to the average degree of the quality of drive. The key is obtained by concatenating the VehicleID with the Start attribute. In Section 6.5, the

Table 6.3: Service class' attributes

Attribute	Type
ServiceID	String
DriverID	String
VehicleID	String
StartService	Timestamp
EndService	Timestamp
Oxygen	Real
MeanVelocity	Real
NumberChanges	Real
MaxPeak	Real
EngineRPM	Real
MeanFuelConsumption	Real
BatteryStatus	String
DriveLevel	Integer

meaning of these data is explained.
In Tables 6.1, 6.2, 6.3 attributes related to each classes are reported.

How does it works?

There are three main phases: Beginning of the Service, During the Service and Conclusion of the Service.
During the first step, the driver proceeds with the authentication by inserting its credentials and the specific VeichleID through an application installed on the smartphone. These data are checked by the Smart Contract which verifies the driver and the vehicle. In case of a positive validation, the Smart Contract creates a new instance of Service. The key, ServiceID, is obtained by concatenating the VehicleID with the Start timestamp. The latter is sent by the driver's application together with the credentials and it is a timestamp which corresponds to the Start of the Service. Thus, the StartService, DriverID and Vehicle ID attributes are set. At this point, the Smart Contract sends to the Vehicle a "Start" message containing the ServiceID information and the second phase begins.
The second step involves the extraction of data during the service. Specifically, the driver can notify the presence of anomalies by using the application. During the service, data related to the driving style are monitored by IoT sensors installed on the vehicle.
Then, there is the third phase related to the Conclusion of the Service. When the driver ends the service, he/she send a "Conclusion" message to the Smart Contract,

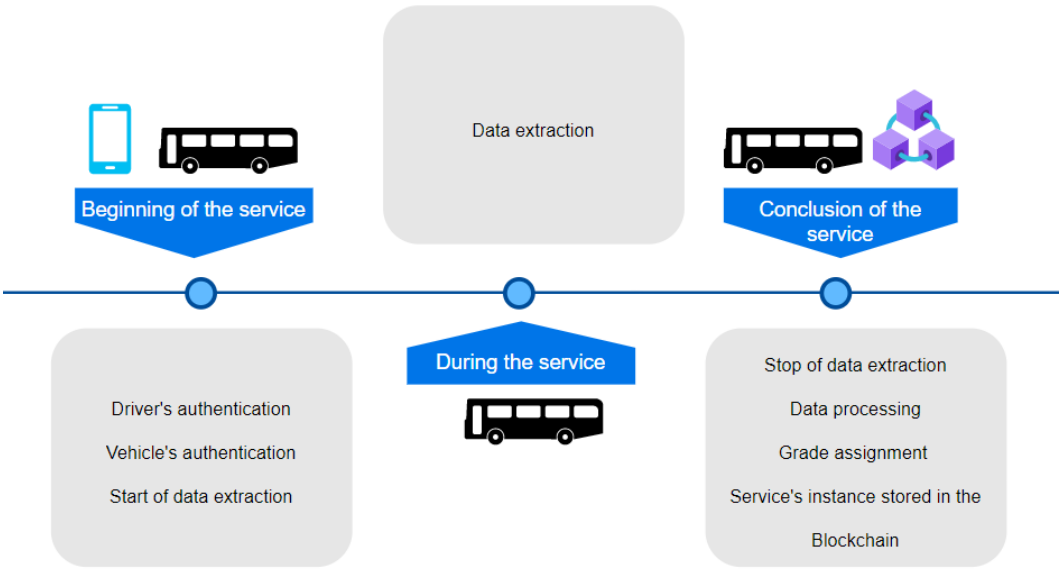

Figure 6.1: The main phases of data analysis.

through the application, with the EndService timestamp. The Smart Contract sets the EndService timestamp to the service instance. Then, it sends to the vehicle a "Conclusion" message and the vehicle replies by sending the processed data of the service. At this point, the Smart Contract sets these data within the service instance. At the end of this step, the service instance is completed and entered to the Blockchain.

6.3 Stored data

In order to perform the desired analysis, the definition of which data to extract is essential.
Data are extracted from the vehicle, specifically, in the considered scenario, from the On Board Diagnostic, OBD II.
From OBD II the extraction of several types of information is possible: for example, it is possible to monitor the speed, the fault codes, the engine RPM...
As written before, the purpose is to store data in the Blockchain and encourage drivers to drive better in order to offer a better service and to reduce costs for the company, too. Thus, it is important to choose data that allow to reach this goal.
For this proposal of application, data that will be considered are the speed, the engine RPM, the O_2 in the exhaust, the load of the vehicle (which corresponds

to the total weight of people on it), fuel consumption, the battery status and the total distance. Specifically, these data are extracted from the vehicle, as it will be explained in Section 6.5.
For what concerns data related to the driver, some considerations are important. Since driver's information involves personal data, in order to be compliant with the GDPR [35], it is important to obtain the consensus of the drivers, who have to be properly informed about how their data are processed and for which purposes. Thus, it is essential to explain how the application works and how data are stored and analysed. The consensus of the drivers is fundamental: if a person does not sign for the consent, his/her data cannot be stored.
In the case in which drivers give their consensus, the useful drivers' data to collect are: the driver's identifier, name, surname, password and the business email address. These extracted data, especially those from the vehicle, are processed in order to obtain information of each service to be stored in the Blockchain.

6.4 Data privacy: Personal data and Driver's involvement

As written in Section 6.3, the project involves the storage of personal data of drivers. The choice of considering personal data, such as name and surname, is made in order to define system which rewards driver proportionally to the quality of their drive.
Therefore, before collecting these personal data, the company has to properly inform drivers about the purpose of this collection and analysis. The adoption of the Blockchain technology can enforce the transparency of data processing: for example, the definition of the grade of the quality of the services and, consequently, the rewards assignments are done through the smart contract whose code is public within the network. Thus, the logic of the system can be known by everybody inside the Blockchain system.
Furthermore, drivers can monitor their services and grades also along time, because Blockchain assures the immutability of stored data in such a way to preserve the historical data.
In the case in which a driver does not give the consensus, the application will not store and extract personal data. In this scenario, the only collected data will be the ones sent by the IoT system in the vehicle.
Another important aspect is related to the smartphone application: drivers should install it on their smartphone but they could be against it. Thus it is essential to consider also the case in which a driver, who gives the consensus for the personal data storage, does not want to install an application on the personal phone. Different solutions can be adopted: the company could give a business smartphone

to the drivers in such a way to not involve personal devices. Another possibility could be installing a smart card reader: drivers can use their smart card to identify themselves, then the reader sends data to the network by means of a sensor which works in the same way with respect to the smartphone application. Then, in order to monitor their services and rewards, drivers can access to their profile by means of the browser application version without the need of installation.
In this thesis, the presented project considers the case in which drivers use a smartphone, but the functioning principles and the working steps are the same also in the case in which the smart card reader's solution is adopted.
Another important point is the choice of not to send personal data to the Raspberry PI: in order to protect data privacy, on the vehicle information relating to the driver are not collected. The only exception is related to the case in which the identification happens via the smart card reader: in this case, the IoT sensor receives the data related to the login but it deletes them when the service ends.

6.5 IoT system

In order to extract data from the vehicle and send them to the Blockchain network, an IoT system is implemented.
There are possible solutions and the choice depends on many factors, such as the type of vehicle and the technology already installed on it. A lot of data are already monitored and extracted from CAN network within the vehicle. Therefore it is possible to read them by using the OBD II. This way implementing other IoT sensors should be redundant.
In this thesis, a possible architecture is suggested, but it might vary depending on the specific scenario and requirements.
A Raspberry PI is installed, in order to monitor and collect data from the different sources. It collects all data of the vehicle, processed them and send to the Smart Contract which will store them to the Blockchain.
An IoT sensor is placed in order to monitor the load of the vehicle.
For what concerns the speed, the engine RPM, the O_2 in the exhaust, fuel consumption, the battery status and the total distance, they are extracted from the CAN network. These data are transmitted to the Raspberry PI through a CAN hat.
Data obtained from the CAN network are extracted every second, so as to monitor them in a quite real-time way. At the end of the service, the Raspberry PI process them and compute some statistics that will be stored in the Blockchain.
The load of the vehicle is sent every 2 minutes, thus with a lower frequency with respect to other data because it is not necessary to extract it in a real-time way.

Figure 6.2: Raspberry Pi with CAN hat.

In the scenario in which the driver did not give the consensus, a signal is sent to the Raspberry PI. This signal is sent when a driver, who does not belong to the Blockchain network, passes the badge to notify the beginning of his, or her, activity. Indeed at the beginning of their shift, drivers have to pass their badge, in order to certify their presence. It is possible to set an additional control which checks if the driver is registered to the Blockchain system. If not, as said before, a signal is sent to the Raspberry PI. This signal does not contain personal data, but it contains just the "StartService" timestamp. The Raspberry PI will send a "Start" message to the Service Smart Contract and the system will work as described before. At the end of the service the driver will pass the badge which acts as before sending an "EndService" timestamp to the Raspberry PI which will send it to the Smart Contract. Thus, the only difference with the smartphone version is that the Raspberry PI coordinates the starting and ending phase by operating as the smartphone but without the personal information related to the driver. In such a scenario, the service instance will have the "DriverID" field equal to "N.A.".
Another important consideration is related to the data processing. In order to avoid the overloaded of the chaincode and of the Blockchain network, all data are processed by the Raspberry PI, which has a sufficient computational power to compute the required analyses. Then, at the end of the service and of the computations, the Raspberry PI will send the result to the Blockchain.

6.6 IoT communication protocol

The installed IoT sensors, which, in the proposed solution, measure the GPS position and the load of the vehicle, communicate with the Raspberry PI through a WiFi network.
For what concerns the communication protocol, the choice is mainly between HTTP and MQTT protocols. In Table 6.4, a brief comparison between these two

communication protocols is presented.
HTTP, HyperText Transfer Protocol, is a protocol well-suited for the creation and transfer of document on Internet. Specifically, through a TCP connection, it supports unidirectional data flow: usually the server just responds to the clients which asks for a service. It is a stateless protocol: it only serves one request at time and the server-client connection is closed when the server finishes to responds to the client. Thus, a new connection is established for each request.
MQTT, Messagge Queuing Telemetry Transport, [36], is designed to work with IoT systems. It allows to establish bidirectional communication and it is also possible to create stateful connections, in which multiple requests of the same client can be stored. MQTT results lightweight, efficient and high scalable. In case of a temporarily unreliable network, this protocol allows to the client to reconnect with the server in a short time.
Thus, the chosen protocol is MQTT, because of its properties.
The Raspberry PI is configured as a MQTT Server, while the IoT sensors are the Clients. In this way, the Raspberry PI is able to receive data from sensors and process them.

6.7 Data processing

When the Raspberry PI, the MQTT server, receives the "Start" message from the Smart Contract, it sends to the IoT sensors and devices installed on the vehicle the request of receiving data. Thus, the IoT sensors start sending their data to the Raspberry PI. These data are sending every second, except for the load of the vehicle which is sent every 2 minutes.
When the Raspberry PI receives the "Conclusion" message from the Smart Contract, it sends a message to the IoT sensors in order to stop the transmission.
At this point extracted data are processed and analysed by the Raspberry PI.
The O_2 in the exhaust is used to monitor the emissions of the vehicle. This

Table 6.4: HTTP vs MQTT

	HTTP	MQTT
Architecture	Client-Server, Request and Response	Client-Server, Publish and Subscribe
Transport	TCP	TCP
Data flow	Unidirectional	Bidirectional
Bandwidth consumption	Heavier	Lighter

parameter is also related to the fuel consumption. For this reason it is useful for the analysis of the pollution produced by the vehicle and to understand if the fuel consumption is too much high. According to the type of vehicle, this value is monitored and for each service, its average is computed and stored in the Blockchain.
As said before (6.2), the proposed version of the project aims to improve the services and minimize costs by reducing the fuel consumption. In order to do so the speed, the engine RPM and the fuel consumption are monitored. Specifically, constant and not too much high velocities and low values of the engine RPM contribute to reduce the fuel consumption.
For this reason, the values of the velocity are analysed in such a way to detect abrupt changes of velocity and peaks. The mean speed, the number of abrupt changes of velocity and the higher peak are computed. Another factor is the load of the vehicle: the average load is computed and taken into account when considering the quality of a drive.
A change of velocity can be defined abrupt according different ways. For example, a change could be considered abrupt if in less than 3 seconds the velocity increases or decreases by 20%.
The battery status and the total distance are useful parameters to monitor. Only their last ones values are stored.
In Table 6.5, a summary of the processed data is reported. These are the data that will be stored in the Blockchain.

Table 6.5: Processed data

Stored Data	Storage purpose
Oxygen	Vehicle Maintenance
MeanVelocity	Reducing the fuel consumption
NumberChanges	Reducing the fuel consumption
MaxPeak	Reducing the fuel consumption
EngineRPM	Reducing the fuel consumption
MeanFuelConsumption	Monitoring the fuel consumption
BatteryStatus	Vehicle Maintenance

6.8 Blockchain Platform: Hyperledger Fabric

After the definition of the IoT system, it is essential to establish which type of Blockchain to choose and the platform to adopt.
Firstly, it is necessary to understand if, in this application, a permissionless network could be better than a permissioned one.

A permissionless and public Blockchain platform leads to a higher lever of decentralization, because everyone can be a node of the chain. Furthermore, it allows to reach full transparency. On the contrary, since everybody can access to data, there is the problem related to data privacy: a company might prefer keeping secret its data and information and protecting them from external sources. This is because data have a value and companies should choose not to share them with other parties. For this reason, a private platform is often preferred, in a business environment.
Another important element to consider is related to the transactions: in order to validate new block a public platform needs to reach an agreement, by means of a consensus protocol such as Pow and PoS; for this reason the validation of transactions results slower and more expensive than the validation of transactions in a private blockchain. Thus, the choice of a public platform for a company's project, is sustainable if data to be inserted are few and do not require a real-time storage. Indeed, in the case in which a lot of transactions have to be inserted, the costs are quite high and the network could be overloaded, thus resulting in an expensive and not efficient system.
Considering all these aspects, for the proposed application, a private and permissioned solution is chosen.
Among the several private Blockchain platforms (such as Corda [37], Ripple [38], ...), the chosen one is Hyperledger Fabric [21]. As already analysed in Section 4.11.1, this platform supports a widely range of applications and it is often chosen by companies which would like to implement the Blockchain technology into their environment. Furthermore, it allows a very flexible implementation of smart contracts which are fundamental in the proposed application.
Hyperledger Fabric results more flexible and more suitable for this project with respect to other platforms, because it is designed for the industry domain. For example, Corda and Ripple are well-suited for banking and financial applications.

6.9 Driver: registration and login

In the following subsections, the processes related to the registration and login of a driver are presented.

6.9.1 Registration of a new Driver

When a new Driver needs to be added to the network, the IT team has to create a new profile. Specifically, in order to manage this task in a quite automatic way, the IT team provides a method which creates a new profile only if the request is sent by a business email address. In this way, the IT team assigns a business email address to the new driver. Then, the driver has to sent the request to be inserted

into the system by using the application installed on the smartphone: by clicking the button "Register", the driver will sent the business email address. Thus, the method defined by the IT team verifies the existence of the email. Thus, in case of a positive result, the method verifies if the user is already registered. If not, it will create the wallet and a SSI (4.5.1) by means of a hash function. The wallet and the SSI are sent by email to the driver who, at this point, has the personal credentials to access to the system.

6.9.2 Login operation

Drivers need to login in order to use the application installed on the smartphone. Specifically, the login phase is necessary at the beginning and conclusion of each service. In addition to this, drivers need to login also in order to see a resume of their services and the current grade of the quality of their drive. Specifically at the conclusion of a service, after the proper verification of the sent credentials, the Smart Contract verifies that the driver corresponds to the one who started that service.

6.10 Vehicle: registration and login

After the presentation of how the registration and login processes are managed for the Driver class, the Vehicle's ones are described.

6.10.1 Registration of a new Vehicle

When there is a new Vehicle, its registration is needed. This can be done only by authorized organization, such as the IT team, in order to register only authorized vehicles. The VehicleID and the type of the motor are inserted. In this way, the new vehicle is added to the network. A SSI is also created in order to protect the access to the information of the vehicle. It is important to notice that, in order to avoid intrusions to the stored data regarding the vehicles, the VehicleID is not the SSI, but the latter is obtained by means of a hash function, which calculates the hash value of the VehicleID.

6.10.2 Validation of the Vehicle

When drivers communicate the beginning of a service, they have to insert their own credentials and the VehicleID. The driver's credentials are checked by the Login method shown in 6.9.2, while the VehicleID is verified by a proper function defined in the Vehicle's Smart Contract which controls the correctness of the identifier of the vehicle. Specifically, this method computes the hash value of the passed

VehicleID and if it effectively corresponds to the SSI of the same VehicleID, the operation is authorized, otherwise an error will be returned.

6.11 Smartphone Application

Drivers who decide to adopt this project need to download the proper application on their smartphone.
The application simplifies the interaction with the system and by using it drivers can also monitor the history of their services and how the quality of their drive evolves along time.
This application should present also a browser version, in order to allow drivers to access to their data from computers or other devices.
The application presents some sections, presented below.

6.11.1 Biographical Data Section

In this section drivers can read their data, such as name, surname and SSI. This section is important because it contains the credentials and the business email, thus drivers can retrieve them if they do not remember themselves.

6.11.2 Services Section

This section contains two main subsections, described below. It is essential because it is the section through which driver sends the "Start" and "End" messages.

Start A New Service Subsection

In this subsection a form is defined. Drivers have to fill in the fields by entering their DriverID (their SSI) and the VehicleID. Because the drivers can not remember all the VehicleIDs, they need to know the specific VehicleID. This can be notified in several ways. For example, a QR code could be placed on the vehicle and drivers can read it through the camera of the smartphone in order to automatically obtain the VehicleID. Since in order to access to the system the DriverID is necessary, even if unauthorized people try to obtain the VehicleID, they would not be able to use it.
After the compilation of the form, drivers can click the button "Start Service", in order to notify the beginning of the service.
In the browser version, this section is not active.

End Of The Service Subsection

This subsection presents the same form of the previous one, but it is already compiled with the information inserted before. Drivers can confirm and send the "Conclusion" message by clicking the button "End Service".
In the browser version, this section is not active.

6.11.3 Services' Resume Section

In this section there are different subsections which allow the visualization of the past services and the obtained grades.

Past Services Subsection

This subsection shows the history of the services of the driver. By selecting a specific service, it is possible to read a summary of the results, its specific grade and, possibly, the major issues of the service. For example, if a driver exceeded the speed limits during the selected service, this issue would be shown.

History of Grades Subsection

Through this subsection, drivers can monitor how their grades evolves during time. They can see them as a list, but also by means of a chart which show how grades evolve among time.

6.11.4 Bonus Section

This section shows the list of achieved bonus.

6.12 Smart contract: Service methods

In the following subsections, methods of the Service's Smart Contract are proposed.

6.12.1 Creation of the Service

Before starting a new service, the driver sends through the smartphone application a "Start" message. Specifically, driver has to insert the personal SSI and the VehicleID. When the driver clicks the button to send this data, a timestamp "StartService" is automatically generated by the application. In this way, the "Start" message contains the driver and vehicle's identifiers and the timestamp of the beginning of the service.
These data are checked, as shown in 6.9.2-6.10.2, and in case of a positive validation,

the new service instance is created.
The identifier of a service, ServiceID, is defined by concatenating the VehicleID string with the timestamp StartService sent by the driver.
Thus, in order to keep trace of the driver of each service, a join is executed: the DriverID is stored in the service instance.
Therefore at this point, the service instance contains the ServiceID, VehicleID, DriverID and the StartService. These data are not already stored in the Blockchain, but momentarily stored by a private database trough which the Smart Contract can store temporarily data which will be used by other functions.
Thus, the Smart Contract sends a message to the vehicle, containing the ServiceID. In this way the IoT system starts to extract and process data.

6.12.2 Conclusion of the Service

At the end of the service, the driver sends an "Conclusion" message to the Service Smart Contract, containing the VehicleID, the DriverID and the "EndService" timestamp. Through these data, a proper method of the Smart Contract obtains the right service instance and verifies the correctness of the credentials. Then, it sends a message to the vehicle, which responds by sending the processed data. By analyzing the processed data, the Smart Contract assigns a grade to the drive. In order to give bonus proportionally to the quality of drive, the definition of classes is presented. Specifically in the proposed project, five degrees of quality have been defined.
Degree 0 corresponds to the base level, drivers belonging to this class do not receive bonus.
With degree 1 drivers receive a little bonus.
By best degree is 4: the quality of drive is excellent and in this case the reward is maximum.
At this point, the Smart Contract contains already all the required data to add to the service instance. Therefore, the service instance is completed and can be stored in the Blockchain system.
It is important to notice that, as written in Section 6.5, in order to avoid the overloaded of the network, data are processed by the IoT system, in such a way to lighten the Smart Contract's methods. The assignment of the grade is done by the Smart Contract in order to achieve a higher transparency of the system. Due to this, drivers are able to know the parameters according to which their drives are evaluated.
When the service is stored in the Blockchain structure, a brief summary of the service is sent to the driver. Furthermore, the driver receives also the grade of the quality of the drive: in this way each driver can monitor it along time. In 6.1, a possible implementation of the creation of a new service is presented. In the

proposed code, the grade has been already assigned by another method, thus it is not computed here, but just passed as the other attributes. Thus, in this method, after the proper control needed to check if the service already exists, the service is created and added to the chain.

Listing 6.1: ChainCode, Example of Java code of method used for the creation of a new Service

```
@Transaction(intent = Transaction.TYPE.SUBMIT)
    public Service CreateService(final Context ctx, final String
    ServiceID, final String VehicleID, final String DriverID, final
    timestamp StartService, final timestamp EndService,final float
    Oxygen,
        final float MeanVelocity, final float NumberChanges, final
    float MaxPeak, final int EngineRPM, final float
    MeanFuelConsumption, final String BatteryStatus, final int
    DriveLevel) {
        ChaincodeStub stub = ctx.getStub();

        if (ServiceExists(ctx, ServiceID)) {
            String errorString = "Service Creation Error: the Service
     with ID =  "+ ServiceID +" already exists!";
            System.out.println(errorString);
            throw new ChaincodeException(errorString,
            ServiceTransferErrors.SERVICE_ALREADY_EXISTS.toString());
        }

        Service service = new Service(ServiceID, VehicleID, DriverID,
     StartService, EndService, Oxygen,
        MeanVelocity, NumberChanges, MaxPeak, EngineRPM,
        MeanFuelConsumption, BatteryStatus, DriveLevel);
        String serviceJson = genson.serialize(service); // in order
    to sort the service
        stub.putStringState(ServiceID, serviceJson);

        return service;
    }
```

6.13 The application software

All the technologies used in the project are connected to each others through an application software. Specifically, it creates the environment within which the IoT system and the Blockchain network operate. In this way, the majority of data are produced and managed internally, without the need of a third party to ensures reliability and trust. Furthermore, it is possible to set the format of data processed by the Raspberry PI and by the smartphone in such a way that they

can be correctly stored in the Blockchain network without the need of oracles.
As it will be shown later, a possible improvement of the project consists on considering also external data such as the traffic information. In this case, these external data sources will be tested by an oracle in order to check the correctness and the format of data.
Therefore, the proposed project considers an internal environment composed of the vehicles IoT system, the Blockchain system involving also the smart contracts, the IT management interface and the user interface (the smartphone application).
The application software allows to obtain such internal environment with all its interfaces and systems leading thus to the connection of all these parts.

6.14 Smart Contract: Reward system

For what the reward system concerns, there are various possibilities. In this proposed project one possible solution is presented and other possible hints will be described later.
Each service has a grade and, as proposed in 6.12.2, there are five possible classes. The overall monthly grade is managed by a smart contract in order to automate the process and to make the system transparent. Specifically, at the end of the month the smart contract is automatically activated and computes the following operations.
On the last day of each month, for each driver the average grade is computed. Furthermore, also the range between the maximum and minimum grades is considered: if this difference is less or equal than 2, the average is rounded up, otherwise it is rounded down.
For example, if a driver has an average grade equal to 2.4, and the maximum

Table 6.6: Overall monthly grade criteria

	Range Max-Min	Average
	> 2	Rounded Down
	≤ 2	Rounded Up

and minimum values are, respectively, 2 and 3, the overall monthly grade will be 3. Instead, if the average grade is 2.4 but the maximum and minimum are 0 and 4, the overall monthly grade will be 2.
In Table 6.6 the above described criteria to describe the overall monthly grade are defined.
The overall monthly grade is then stored in the profile of each driver. These grades are also sent to the IT team that has to assign the corresponding bonus to drivers.

Drivers can check the correctness of the bonus by comparing the overall grade received through the application with the one sent by the IT team together with the bonus. If the IT team sends a different degree, drivers can notify the error.
The bonus corresponds to vouchers that drivers can use in supermarkets and their values are established in a proportional way according to the overall monthly grade.

6.15 Some future possible improvements

In this Section, some possible improvements of the project are presented.
One of the most critical aspects is related to the reward system implementation. In fact, more the algorithm is accurate and more the monitoring of the quality of the drives is efficient. Thus, as second step, an improvement of the reward system can be taken into account.
In the proposed version, the focus is on the improvement of the service by encouraging drivers to drive better avoiding, where possible, abrupt changes of velocity, thus obtaining also a reduction of the fuel consumption.
A possible improvement could be considering the traffic conditions. In this way, a better analysis can be performed because also the external environment is analysed and it is possible to define if drivers optimize the velocity. For example, in case of heavy traffic, velocities will be lower, whereas in case of light traffic, it is reasonable to expect higher velocities, in order to avoid delays.
In order to analyse traffic conditions, the implementation of an oracle is necessary.

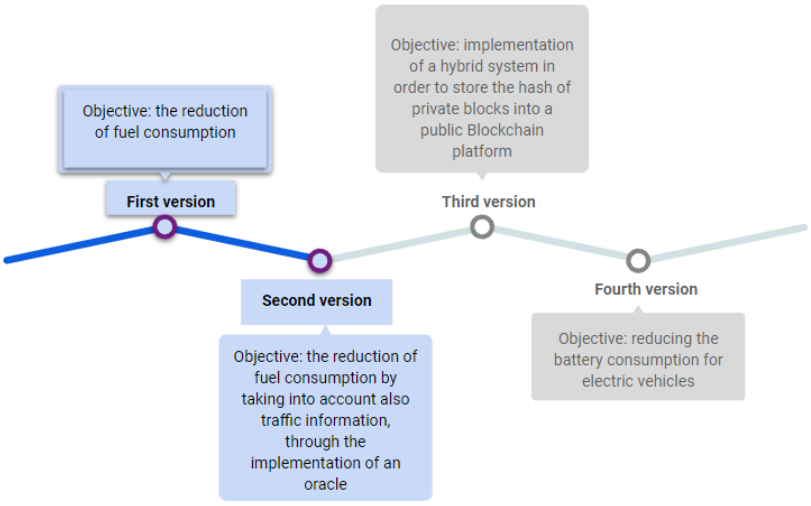

Figure 6.3: A possible road map of future improvements.

This is because the traffic information are external data and in order to guarantee the trustworthiness of the source (the web, for example), an oracle has to be implemented. Furthermore, the GPS position will be monitored by an IoT sensor in order to capture only traffic data related to the area nearby the vehicle.
Another possible improvement regards the choice of the bonus. In the proposed version, only vouchers for the supermarket are considered. In the case in which the system works well and there is a high participation of the drivers, the extension of the kinds of bonus could be considered. In this case in the Bonus section of the smartphone application an additional subsection drivers could choose the type of bonus they preferred such as ticket restaurant, cinema ticket and vouchers to use in bookshop.

Another factor to consider for the a future version of the project includes the increasing involvement of electric vehicles. In this scenario, the algorithm should consider different parameters, since the objective is no longer the reduction of the fuel consumption, but the optimization of charges, in order to save time and resources.

Chapter 7

Conclusions

This thesis focuses on the integration of IoT and Blockchain technologies.
In order to better understand how it is possible to improve this integration by overcoming some of the limitations of both technologies, an in-depth study of their features has been conducted.
The connectivity of IoT systems allows to implement these devices also in scenarios in which before it was not possible to obtain a connected environment. As shown, thanks to their small dimensions, IoT sensors can be applied in places with limited space availability. New solutions able to monitor and manage multiple sensors have been defined, such as Raspberry PI, which results very flexible thanks to its small dimensions and features. It is an useful small-board computer and it allows to store and analyse data extracted by different sensors.

IoT devices often collect personal data, thus, a high level of security should be guaranteed. Unfortunately, there are several malicious attacks which can affect IoT sensors and systems, therefore the security is still a critical aspect of this technology. By using the Blockchain, security measures are implemented in a proper way. Indeed, even if the same security measures can be defined in other storage solutions, they are strongly integrated into the Blockchain structure, whose functionality is entirely based on them. On the contrary, in other types of databases, they should be added as additional measures.

Another challenging aspect is related to data integrity: with respect to other storage solutions, data immutability is guaranteed by the Blockchain system: in this way, historical data can be observed and analysed in a trustworthy manner. This feature is essential in applications similar to the one proposed in this thesis: the company of public transport can rely on trusted historical data in order to better compare the targets of the analysis such as driver's behaviour and fuel consumption.

On the other hand, Blockchain also presents some critical weaknesses to take into account, too. One of the most important in IoT context is the difficulty of Blockchain of handling big data: this is a critical limit because in a lot of cases IoT sensors extracts a huge volume of real-time data.

The choice of using a private Blockchain is due to the company's requirements. By adopting a private platform, some of the advantages of a public one could be lost, such as the high level of decentralization. In fact, Blockchain was born as a public system and designed for financial transactions. Among the years, due to its evolution, Blockchain began to be also adopted for other types of applications, such as automotive, medical and smart cities. Thus, in a lot of cases in these different scenarios, the companies' requirements, such as those related to data privacy with respect to the external environment, lead to the definition of new private and permissioned platforms, such as Hyperledger Fabric. A possible way to preserve high levels of decentralization could consists on adopting hybrid systems. Furthermore, the choice of using a public platform leads to additional costs related to the fees: thus, storing all the results on a public Blockchain such as Bitcoin should be unfeasible in terms of costs.

The proposed project takes into account all these aspects. Since the aim is to analyse the feasibility of applications involving the integration of IoT and Blockchain, and because of the early stage of the studies of this integration, this thesis presents a first version of the project. Possible improvements are mainly related to the reward system which could consider also different factors such as the presence of accidents which have an impact on the service. Therefore, this version shows how to connect and use the essential devices and tools required for the application. For example, a way to avoid the overloaded of the Blockchain network and to overcome the problem related to the collection of big data is provided: only the results of the analysis are sent to the Blockchain. In this way it is possible to obtain a feasible system both by preserving the capacity of IoT sensors of processing big data and by storing only the results on the Blockchain.

Even if this project proposes how to mitigate some of the critical aspects of IoT and Blockchain technologies, such as the big data, there are still some challenges: the next step could be the adoption of a hybrid solution. In fact, after the storage on the private platform, the hash value of the blocks could be stored on a public Blockchain, such as Bitcoin.

Appendix A

Chaincode Go script: Definition and creation of Asset and Ledger's initialization

Below, a script written in Go are reported. They are related to the definition and creation of Asset and the initialization of the ledger. This script is provided by the Hyperledger Fabric documentation [39].

```
    type Asset struct {
    AppraisedValue int     `json:"AppraisedValue"`
    Color          string  `json:"Color"`
    ID             string  `json:"ID"`
    Owner          string  `json:"Owner"`
    Size           int     `json:"Size"`
}

// InitLedger adds a base set of assets to the ledger
func (s *SmartContract) InitLedger(ctx contractapi.
    TransactionContextInterface) error {
    assets := []Asset{
        {ID: "asset1", Color: "blue", Size: 5, Owner: "Tomoko",
    AppraisedValue: 300},
        {ID: "asset2", Color: "red", Size: 5, Owner: "Brad",
    AppraisedValue: 400},
        {ID: "asset3", Color: "green", Size: 10, Owner: "Jin Soo",
    AppraisedValue: 500},
```

```
15          {ID: "asset4", Color: "yellow", Size: 10, Owner: "Max",
    AppraisedValue: 600},
16          {ID: "asset5", Color: "black", Size: 15, Owner: "Adriana",
    AppraisedValue: 700},
17          {ID: "asset6", Color: "white", Size: 15, Owner: "Michel",
    AppraisedValue: 800},
18      }
19
20      for _, asset := range assets {
21          assetJSON, err := json.Marshal(asset)
22          if err != nil {
23              return err
24          }
25
26          err = ctx.GetStub().PutState(asset.ID, assetJSON)
27          if err != nil {
28              return fmt.Errorf("failed to put to world state. %v", err
    )
29          }
30      }
31
32      return nil
33  }
34
35  // CreateAsset issues a new asset to the world state with given
        details.
36  func (s *SmartContract) CreateAsset(ctx contractapi.
        TransactionContextInterface, id string, color string, size int,
        owner string, appraisedValue int) error {
37      exists, err := s.AssetExists(ctx, id)
38      if err != nil {
39          return err
40      }
41      if exists {
42          return fmt.Errorf("the asset %s already exists", id)
43      }
44
45      asset := Asset{
46          ID:             id,
47          Color:          color,
48          Size:           size,
49          Owner:          owner,
50          AppraisedValue: appraisedValue,
51      }
52      assetJSON, err := json.Marshal(asset)
53      if err != nil {
54          return err
55      }
56
```

```
57      return ctx.GetStub().PutState(id, assetJSON)
58  }
```

Appendix B

Cryptocurrencies: some statistics

In the following image, the top 10 cryptocurrencies ordered by Market Cap are reported. These values are provided by CoinMarketCap [40] and they refer to June 14, 2023.

Watchlist | Portfolio | Cryptocurrencies | Categories | Memes | Bitcoin Ecosystem | Liquid Staking Derivatives | Metaverse | Show rows 100 | Filters | Customize

#	Name	Price	1h %	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 Days
1	Bitcoin BTC	$26,006.34	0.09%	0.03%	2.47%	$504,574,712,758	$10,030,454,144 385,599 BTC	19,401,987 BTC	
2	Ethereum ETH	$1,742.38	0.04%	0.32%	6.19%	$209,476,174,411	$4,247,563,644 2,438,624 ETH	120,224,091 ETH	
3	Tether USDT	$0.9999	0.02%	0.00%	0.06%	$83,484,019,419	$16,575,704,711 16,576,328,093 USDT	83,495,780,838 USDT	
4	BNB BNB	$246.94	0.19%	2.89%	7.09%	$38,486,195,653	$735,652,332 2,987,978 BNB	155,854,075 BNB	
5	USD Coin USDC	$1.00	0.02%	0.01%	0.00%	$28,148,575,645	$2,378,561,075 2,378,431,037 USDC	28,168,184,455 USDC	
6	XRP XRP	$0.5058	0.07%	4.51%	4.33%	$26,293,603,411	$1,573,095,905 3,116,151,018 XRP	51,987,017,573 XRP	
7	Cardano ADA	$0.2747	0.12%	0.57%	17.40%	$9,591,875,657	$199,907,259 730,173,245 ADA	34,914,604,763 ADA	
8	Dogecoin DOGE	$0.06187	0.02%	0.39%	8.57%	$8,647,473,884	$149,801,918 2,424,221,414 DOGE	139,763,556,384 DOGE	
9	TRON TRX	$0.07253	0.07%	1.04%	6.15%	$6,533,845,130	$138,496,751 1,911,561,862 TRX	90,084,644,855 TRX	
10	Polygon MATIC	$0.6534	0.29%	0.79%	16.11%	$6,069,383,722	$291,656,352 447,749,269 MATIC	9,289,469,069 MATIC	

Bibliography

[1] URL: https://www.turintech.it/home (cit. on p. 1).

[2] Stephen Ornes. «The Internet of Things and the explosion of interconnectivity». In: *Proceedings of the National Academy of Sciences* 113.40 (2016), pp. 11059–11060 (cit. on p. 3).

[3] Kevin Ashton et al. «That 'internet of things' thing». In: *RFID journal* 22.7 (2009), pp. 97–114 (cit. on p. 3).

[4] URL: https://www.01net.it/iot-europa-96-miliardi-2023/ (cit. on p. 4).

[5] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. «A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling». In: *ACM Computing Surveys (CSUR)* 53.1 (2020), pp. 1–32 (cit. on pp. 4, 5, 28, 29, 37, 38).

[6] URL: https://www.raspberrypi.com/ (cit. on pp. 6, 7).

[7] URL: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307 (cit. on p. 8).

[8] Evangelos Theodoridis, Georgios Mylonas, and Ioannis Chatzigiannakis. «Developing an iot smart city framework». In: *IISA 2013*. IEEE. 2013, pp. 1–6 (cit. on p. 8).

[9] Nilufar Neyestani, Maziar Yazdani Damavandi, Miadreza Shafie-khah, and João P. S. Catalão. «Modeling the PEV traffic pattern in an urban environment with parking lots and charging stations». In: *2015 IEEE Eindhoven PowerTech*. 2015, pp. 1–6. DOI: 10.1109/PTC.2015.7232637 (cit. on p. 9).

[10] M Mazhar Rathore, Awais Ahmad, Anand Paul, and Seungmin Rho. «Urban planning and building smart cities based on the internet of things using big data analytics». In: *Computer networks* 101 (2016), pp. 63–80 (cit. on p. 9).

[11] R Dhaya and R Kanthavel. «A wireless collision detection on transmission poles through IoT technology». In: *Journal of trends in Computer Science and Smart technology (TCSST)* 2.03 (2020), pp. 165–172 (cit. on p. 9).

[12] Eric A Brewer. «Towards robust distributed systems». In: *PODC*. Vol. 7. 10.1145. Portland, OR. 2000, pp. 343477–343502 (cit. on p. 12).

[13] Stuart Haber and W Scott Stornetta. *How to time-stamp a digital document*. Springer, 1991 (cit. on p. 15).

[14] Satoshi Nakamoto. «Bitcoin: A peer-to-peer electronic cash system». In: *Decentralized business review* (2008), p. 21260 (cit. on pp. 15, 19, 21, 25).

[15] URL: `https://blog.osservatori.net/it_it/self-sovereign-identity-spiegazione-applicazioni` (cit. on p. 18).

[16] Gavin Wood et al. «Ethereum: A secure decentralised generalised transaction ledger». In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32 (cit. on p. 19).

[17] Sergio Demian Lerner. *DagCoin: A Cryptocurrency Without Blocks*. 2015. URL: `https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf` (cit. on p. 21).

[18] Marc Pilkington. «Blockchain technology: principles and applications». In: *Research handbook on digital transformations*. Edward Elgar Publishing, 2016, pp. 225–253 (cit. on p. 22).

[19] Nick Szabo. «Formalizing and securing relationships on public networks». In: *First monday* (1997) (cit. on p. 22).

[20] URL: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode.html` (cit. on pp. 22, 27).

[21] URL: `https://hyperledger-fabric.readthedocs.io/en/release-2.5/` (cit. on pp. 22, 26, 52).

[22] URL: `https://chain.link/` (cit. on pp. 24, 25).

[23] URL: `https://docs.chain.link/vrf/v2/introduction` (cit. on p. 24).

[24] URL: `https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf` (cit. on p. 26).

[25] URL: `https://www.ilsole24ore.com/art/blockchain-ecco-smart-city-giapponesi-progettate-catena-blocchi-AEjJxqrE?refresh_ce=1` (cit. on p. 38).

[26] URL: `https://www.fujitsu.com/global/about/resources/news/press-releases/2018/0514-02.html` (cit. on pp. 38, 39).

[27] HS Jennath, S Adarsh, Nikhil V Chandran, R Ananthan, A Sabir, and S Asharaf. «Parkchain: a blockchain powered parking solution for smart cities». In: *Frontiers in Blockchain* 2 (2019), p. 6 (cit. on pp. 39, 40).

[28] URL: `https://swachhcoin.com/` (cit. on p. 40).

[29] *Swachh Coin White Paper.* URL: https://swachhcoin.com/whitepaper.pdf (cit. on p. 40).

[30] URL: https://driveindi.com/ (cit. on p. 41).

[31] URL: https://www.linkedin.com/company/driveindi/ (cit. on p. 41).

[32] URL: https://driveindi.com/a-futuristic-vision-for-car-owners-2/ (cit. on p. 41).

[33] URL: https://www.bmw.it/ (cit. on p. 41).

[34] URL: https://www.bmw.com/it/innovation/blockchain-automotive.html (cit. on p. 41).

[35] URL: https://www.garanteprivacy.it/regolamentoue (cit. on p. 47).

[36] URL: https://mqtt.org/ (cit. on p. 50).

[37] URL: https://corda.net/ (cit. on p. 52).

[38] URL: https://ripple.com/ (cit. on p. 52).

[39] URL: https://github.com/hyperledger (cit. on p. 63).

[40] URL: https://coinmarketcap.com/ (cit. on p. 66).