Universidade do Minho
Escola de Engenharia
Departamento de Informática

Carlos Daniel Martins da Costa

Continuous monitoring of door-to-door postal service

Dissertação de Mestrado
Mestrado em Engenharia Informática

January 2020

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Carlos Daniel Martins da Costa

Continuous monitoring of door-to-door postal service

Dissertação de Mestrado
Mestrado em Engenharia Informática

Master dissertation
Master Degree in Computer Science

Dissertation supervised by
Pedro Rangel Henriques
Nuno Feixa Rodrigues

January 2020

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Carlos Costa

Assinado por : **Carlos Daniel Martins da Costa**
Num. de Identificação: BI14221621
Data: 2020.07.26 15:54:09+01'00'

CHAVE MÓVEL

ABSTRACT

Logistics services, including express mail delivery areas, have been growing significantly by the increase in the volume of e-commerce activity worldwide. It is expected that the rise in the level of digital competencies of companies and citizens will not only promote considerable growth in this sector over the next few years, but also demand higher levels of efficiency, quality, and modernization of digital platforms for interaction with customers.

In terms of continuous monitoring, new technologies offer potential, namely the use of GPS devices to collating coordinates. With system integration, the collected coordinates can be temporarily saved and then sent to a remote server.

Door-to-door service requires exact locations, so there are certain technologies, which allow us to collect that information accurately without the minimum margin of error. In the context of door-to-door distribution, most companies have simple technology that provides a piece of insufficient information regarding the status of their order, they only present information that the postal service may be delivered, refused, or the addressee may not be found.

Regarding door-to-door distribution, technologies can be implemented to improve the current industry solutions, providing more detailed information about the order status. Thus, a solution was developed based on international standards, that allow live tracking application ensuring also data security through blockchain technologies.

Keywords: Logistics, E-commerce, System integration, Live tracking, Blockchain.

RESUMO

Os serviços de logística, onde se inserem as áreas de entrega expresso de correspondência postal, têm vindo a crescer significativamente, muito impulsionados pelo aumento do volume da atividade e-commerce em todo o mundo. Perspetiva-se que o aumento do nível de competências digitais das empresas e cidadãos venha não só a promover um forte crescimento deste setor nos próximos anos, como também a exigir níveis mais elevados de eficiência, qualidade, e modernização das plataformas digitais de interação com os clientes.

No que se enquadra a uma monitorização continua, as novas tecnologias oferecem potencialidades nomeadamente o uso de dispositivos GPS, para o efeito de recolha de coordenadas. Com a integração de sistemas, as coordenadas recolhidas podem ser guardadas temporariamente e de seguida enviadas para um servidor remoto.

Como se pretende um serviço de porta-a-porta a localização atual deve de ser exata, por isso existem determinadas tecnologias, que nos permitem recolher essa informação de maneira precisa sem que haja uma mínima margem de erro. No que concerne à distribuição porta a porta atualmente a informação apresentada ao utilizador proporciona apenas simples mensagens acerca do estado atual tais como: Entregue, recusado, ou destinatário não encontrado.

De acordo com a distribuição porta-a-porta, tecnologias podem ser implementadas de modo a melhorar as soluções da indústria, providenciando informação mais detalhada acerca do estado da encomenda. Assim, foi desenvolvida uma solução baseada em standards internacionais, que permitem um rastreamento em tempo real assegurando também a segurança dos dados através da tecnologia blockchain.

Palavras-chave :Logística, E-commerce, Integração de sistemas, Localização em tempo real.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

A

AVL Automatic Vehicle Location.

B

BFT Byzantine Fault Tolerance.

BITA Blockchain in Transport Alliance.

BLL Business Logic Layer.

C

CA Certificate Authority.

CFT Crash Fault Tolerance.

D

DAL Data Access Layer.

DGPS Global Positioning System Differential.

DLT Distributed Ledger Technology.

DR Dead Reckoning.

E

ECDSA Elliptic Curve Digital Signature Algorithm.

EDI Electronic Data Interchange.

EDP Energias de Portugal.

ERP Enterprise Resource Planning.

EVM Etherum Virtual Machine.

F

FTL Full truck load.

G

GDPR General Data Protection Regulation.

GPS Global Positioning System.

GSM Global System for Mobile.

H

HF Hyperledger Fabric.

HLF Hyperledger Fabric.

I

IOT Internet-of-things.

ITS Intelligent Transportation System.

J

JSON JavaScript Object Notation.

JWT JSON Web Token.

L

LTL Less than truck load.

M

MSP Membership Service Provider.

MVC Model-View-Controller.

P

P2P Peer-to-peer.

PBFT Practical Byzantine Fault Tolerance.

PDC Private Data Collections.

PDOS Private Data Objects.

PKI Public Key Interface.

POET Proof of Elapsed Time.

POS Proof of Stake.

POW Proof-of-Work.

R

RFID Radio Frequency Identification.

RW Read and Write.

S

SCM Supply Chain Management.

T

TMS Transport Management System.

U

UI User Interface.

W

WMS Warehouse Management System.

1

INTRODUCTION

Nowadays, transportation systems represent a fundamental component in a company success, and with technological evolution, there are many solutions that propose active monitoring of several products.

Customer displeasure and enterprise feedback is an essential factor, and this refers to the efficiency and management in logistics. Delays on delivery, the correct product at an exact time, or even the factor we do not know the order status, are a determining factor for customer satisfaction.

According to European Commission (2019), the transport sector is expected to grow 60% until 2050, being also a counterpart for CO_2 emissions. The European Commission invested around 3 million euros in crucial transport projects to withstand connected mobility in Europe.

The *Intelligent Transportation System (ITS)* came to optimize processes such as improving routes, optimize fuel consumption, among many other tactics. These systems already integrate several tools to help drivers carry out their tasks.

Methods already built before, such as *Automatic Vehicle Location (AVL)*, offer many advantages and benefits since it is the method to determine a vehicle location with *Global Positioning System (GPS)*, however, there are cheaper solutions, and practical techniques to bypass the fact of having to install a device for each vehicle.

The AVL communicates with the base stations or servers through *Global System for Mobile (GSM)* card, but we have a locator always present in our hands due to technological developments.

Thus, in this dissertation, a real-time application is required, which will be focused on web and mobile devices. The simple fact of building a system that supports logistics systems and transportation can contribute to the management and financial status of a company. The mobile devices are very present in the market today, and the fact that it will be transmitted in real-time the door-to-door location leaves the consumer with less anxiety and manage the time more accurately.

The steady growth of technology allows us to build systems dedicated to our needs. One of the main objectives is to ensure customer satisfaction by providing detailed information about the order demonstrating the entire process, being able to prove the guarantee of the authenticity of the product.

The mobile devices have limited autonomy, so running a process or service in the background, optimizes much battery, not being necessary to be connected to a power source. With techniques of software engineering, algorithms that improve procedures for using hardware will be optimized.

Based on this, the main objective is developing a system that it is designed to, ensure authenticity, security, innovation, and accessibility. Software engineering methods will be used to model the system, using use cases, class diagrams, and sequence diagram allowing a broader view of the system in question.

1.1 MOTIVATION AND GOALS

With the evolution of technology and market growth, the companies were obliged to invest technologically in new systems to facilitate management and optimize costs.

As it has previously been seen, the technology AVL present in the market involves a high cost and maintenance, due to the communication and data storage services provided. Some alternatives support this technology, as well as smartphone devices. The access to this smallest equipment is an essential part of today's world. However, the potential that offers and ease of use simplifies the entire process and costs of a company.

A mobile application with a previous study achieves good accuracy, and provides an excellent low-cost option for a company. The application of methods for development requires many pieces of knowledge about security and mobile architecture operation mode. The development of a mobile application based on the client-server model is the most common methodology used by companies, that will be present for many years. Not contradicting that, and being this a correct methodology, this dissertation aims at building something innovative that guarantees the customer's confidence and satisfying the needs of the market.

The main goal of this dissertation is to develop a monitoring system door-to-door for a company to be able to monitor orders without there being any need to purchase types of equipment. Based on the AVL system, it is intended to develop a system able to offer a lower-cost technology, eliminating service annuity for one or more companies from various sectors of logistics.

The guarantee that an order is delivered on time, and their authenticity allows the customer to gain confidence in the company, allowing better market performance compared to other companies, with these aspects, some requirements are fundamental for the operation and monitoring of the door-to-door postal service, by applying active monitoring of vehicles and verify how many vehicles exist, gross weights that can carry, order the load, type of product to be transported, and dimensions.

A web-based mobile application that supports these functions requires processing power, high network availability, and autonomy, due to client-server requests. Components and optimization of code are essential for battery optimization, through the use of network technologies such as mobile data and location services GPS.

According to the market, it is necessary to adapt techniques and studies relative to the developed code, as the basis will be a mobile phone, it is essential to know how architecture works and its components.

The development of this application aims at:

- Developing a system that allows enterprises to track all vehicles.
- Developing a back-end that allows an enterprise to add vehicles/drivers, types of loads.
- Developing a specific mobile application for drivers, with the purpose of sending the location in real time.
- Developing a real-time application that allows the customer to track your order.
- System integration between mobile application and back-end.

These main goals may be adapted due to the needs of the companies, or to improve the system efficiency according to the feedback from customers.

1.2 Dissertation structure

The structure of this dissertation is divided into six parts, organized as it follows:

Chapter 2, the background, is an overview of how supply chains and logistics work, the various types of technologies that exist in this sector, and an introduction to Blockchain technologies and its components.

Chapter 3, presents the state of the art of blockchain in logistics. It will be described how the system works, and what are the main challenges and main flaws to overcome. A standards review and a market analysis are made in order to find the best, or most adequate, technology to develop this Master's project.

In chapter 4, the desired system is modelled and a solution is proposed. The software models are presented in order to specify system details and iterations.

In chapter 5, the architectures that were studied in the background and state of the art will be shown and explained in detail. A comparative table of available technologies was created in order to understand what is the best technology to be adopted for the implementation. Then, a system description is made, and the main events that occur in the system are presented. The setup of the environment is explained, and some code excerpts will be presented. Finally, results from transactions and relevant aspects about the system performance either its Web version, or mobile application will be exhibited.

Finally, in Chapter 6, conclusions are drawn and the obstacles that rose up during the study and implementation are discussed. An analysis is made regarding the system improvements that should be made as future work.

2

BACKGROUND

In this chapter, some aspects of logistics systems will be discussed. Nowadays, almost every distribution company, uses automatic location equipment, using an AVL, in order to optimize costs, and minimize the risk of stealing and fraud during distribution.

However, some systems are expensive, which compromise the adoption and do not offer data confidence since they can be easily manipulated, not satisfying the company's needs.

This chapter of the dissertation delves into the understanding of how logistic systems work and find new methods in order to introduce a new technology inside the business by providing confidence of all involved.

Accordingly, the present sections demonstrate how logistics companies work in general to ensure effectiveness and responsiveness in operations.

2.1 PHYSICAL DISTRIBUTION (ORDERS)

The physical distributions are inserted in the logistics chain, and according to Bertaglia (2003), the products and materials are moved along the supply chain.

The companies receive all the raw material and then transform it into the final product, followed by suppliers for the distribution and finally to customers. However, all companies have an established model for different types of delivery.

It is important to mention that distribution process has a high cost, and it needs much attention so that the end customer does not get impaired. This cost is related to the maintenance of vehicles and the optimization of routes. As it will be described in Figure 1 , the process of Supply Chain demonstrates the complexity of cadence and helps to understand the basics of physical distribution.

The physical distribution aims to deliver the right products to the right recipients in order to obtain the lowest possible cost. Usually, all logistics centers have a distribution center near motorways allowing access to large transports.

In conclusion, we have, as the starting point, physical distribution, aiming that items must reach the destination, with the right conditions, and with competitive prices.

Included:

- Transportation
- Storage
- Material Handling
- Packaging and Protection
- Stock Control

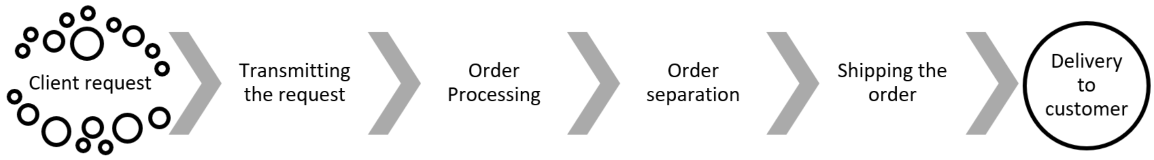

Figure 1: Supply Chain, Adapted from: David J. Clos (2001)

2.2 ROAD TRANSPORT IN LOGISTICS DISTRIBUTION AND ITS MONITORING

Transport in logistics offers several types of transportation such as air, rail, road, and maritime, although the intended and the one that is considered in this dissertation is road transport.

According to Eurostat (2019), the road transport represents a superior performance in the transport of goods compared to the other three types that exist inland, according to the graph presented in the document, road freight transportation represents about 76% compared to the other inland methods.

A fixed cost is presented when transporting passengers or goods, including more flexibility and efficiency due to the facilitating communication between different spaces. However, in contrast, the load capacity may be limited as being profitable for short and medium distance courses.

Road transport offers a point-to-point service, which is used between two points: origin and destination, allowing great functional dispatch and access to remote destinations. Nevertheless, these transports are used by light industries, where road freight transport is fast in small quantities.

It is considered that road transport can carry a wide range of products respecting their limitations, and ordination due to size and length.Godinho (2007). The variation in transport time may be affected by road conditions, weather conditions, enormous traffic, and by the route chosen, being sometimes difficult due to unexpected situations, unable to predict the time by carrier.

Some advantages and disadvantages will be presented below concerning road transport:

Advantages:

- Delivery in distant locations and difficult access.
- Fast delivery in short distance.
- High flexibility of itineraries.
- Good carrying capacity.
- Door-to-door transportation.
- Evolution to become faster and more convenient.

Disadvantages:

- Delays in deliveries (due to unforeseen conditions).
- More vulnerable to robberies.
- High cost of infrastructure.
- Helps to cause traffic jams.
- High consumption of fossil fuels.
- High accident rate.

2.3 TRANSPORT NETWORKS

The transport networks were created to optimize the transportation process. There are several projects of transport networks, illustrating the process of moving products.

Older models cannot go along the expected levels of service due to the effect of market growth that makes logistics a key point for the customer. The customer typically requires one product with the lowest price possible. Inventory and stock provided to individual companies achieve a competitive price in the market. However, we must remember the transport cost, storage quantities, and survey the market so that there is no price disparity.

Previously, the direct delivery network established was "origin to destination", used for the transportation of complete loads *Full truck load (FTL)*. Partial load *Less than truck load (LTL)* means when a truck is not complete. According to Chopra and Meindl (2013), there are several types of transport networks, namely: Cross-docking, Milk-run, and Merge-docking, which are most commonly used to create a distribution transport network or to introduce distribution centers or transit-point centers. The direct network consists of the delivery of a direct mode to the supplier, not having intermediate deposits, simplifying all system operation.

The Milk-run direct transport network is an evolution of the direct delivery, intending to collect all products from the various suppliers and deliver them to a single destination, not requiring intermediate deposits Chopra and Meindl (2013). A large carrier commonly uses a distribution center using cross-docking.

The operation of the network consists of : The Distribution Center which receives load from several customers and manage the stock-based in *Warehouse Management System (WMS)*, preparing the load for distribution. For loads completing a FTL the distribution is direct, and if the cargo is small, are generally distributed by regions.David J. Clos (2001)

All of the networks above are suited to different types of distribution companies since the most adequate alternative for each circumstance is calculated.

Type of network	Benefits	Disadvantages
Direct Delivery	Easy coordination without warehouse	Large-scale inventories
Direct Delivery (Milk-run)	Low shipping cost Low stock	Increasing the complexity of the route
Delivery via Distribution Center (Milk-run)	Low cost for small lots	More complex
CD delivery (Cross-docking)	Very little stock. Reduced transportation cost	More complex in terms of information management

Table 1: Benefits and disadvantages of types of network

The advantages and disadvantages, in table 1 are appropriate to the type of company.

2.4 VEHICLE ROUTING

The distribution of products and logistics are the principal activities of a company, becoming a success in the process of distribution to customers, although their transportation must be performed to meet factors such as time, cost, and efficiency.

Proper planning of activities creates conditions to meet the efficiency and reliability of the service provided by the company, guaranteeing customer satisfaction, and reducing costs Botelho (2003). Vehicle routing is a process that permits the minimization of the distribution costs of a fleet and characterized by the creation of road maps, with stopping points for more effective delivery.

The knowledge of a transport route allows greater security for the load and vehicle due to the possibility of changing the route if it exists a lot of traffic in a particular area or warns about road conditions causing a delay in deliveries. There are several market solutions for route planning, systems that offer almost real solutions can bring many benefits to a company by estimating delivery time. These solutions offer advantages such as: Calculating the best route to be used, optimizing the use of vehicles, economy, sequencing of deliveries proposals through the WMS, increase productivity and calculate delivery time by each customer.

The problem of Vehicle Routing, according to Lásara Fabrícia Rodrigues (2006), is over many years studied and is used to find a set of routes for vehicles that leaves for warehouse distribution. Several customers are forming each route, and a vehicle only visits a customer one time. Each delivery point has a knowledge about the load, and weight cannot exceed the capacity of the vehicle not to commit illegalities. The development of software and hardware allowed the business sectors to generate a reasonable expectation of benefits, with most companies in the industrial sector containing techniques to solve the routing problem.

2.5 TRANSPORT MANAGEMENT SYSTEM

TMS is a subset of *Supply Chain Management (SCM)* and aims to plan, execute, and optimize orders. It is a platform that allows users to manage and optimize daily operations and transportation routes.

This system can be incorporated into *Enterprise Resource Planning (ERP)*, which is already well-known, making it easier to view and control logistics operations.

In view of the market growth and resource optimization, some companies were forced to add TMS. Marques (2002) defines that transport management is the essential part of a logistics system, responsible for raw material and the product movement between all points of a logistics chain. However, due to many assets, this management becomes much more complicated. Figure 2 is an example of TMS adaptation in a company.

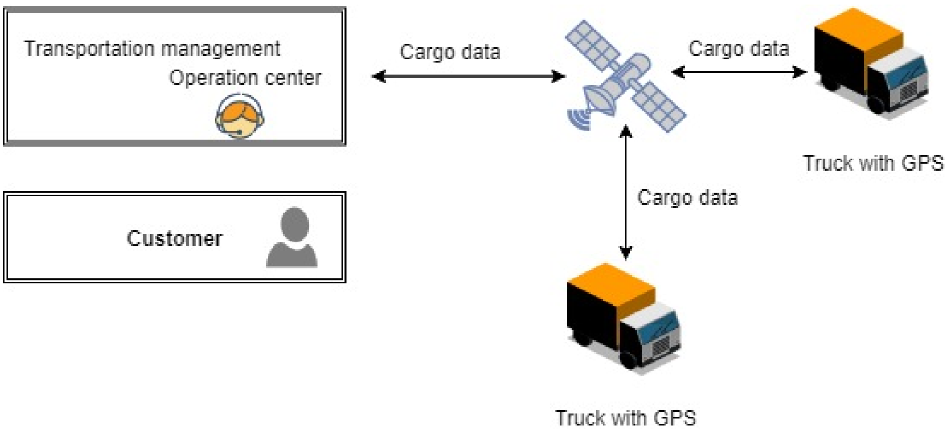

Figure 2: GPS-based TMS Adapted from: WE (2019)

There are several functionalities that TMS system offers, and it is adopted to the best method for each type of business. Marques (2002) divided them into three groups: Monitoring and Control; Negotiation and load control; Planning and execution.

- Monitoring and Control: Making budgets helps control costs (cost per ton), amounts paid for a route, and other costs. It can also be monitored charging and discharging time, customizing the system according to the customer, allows compliance with legislation imposed by market regulators such as fuels, vehicle maintenance, vehicle registration, documentation.
- Negotiation and load control: It is required to save all the route costs to compare later and help in the audits processes. The system must compare the price calculated and the price charged to the customer and check the differences to make a better optimization eventually. Charge control is an essential procedure because they can raise the transported load volume, and yet increase safety. The financial conditions must be stored in the system such as the fraction per load: travel cost, type of vehicle, the route that was made, size of load and destination, transport teams.
- Planning and execution: Some systems can calculate and save time when the vehicle is stopped, whether in transit or discharge, to keep the necessary documentation and check the availability of the vehicles. The routing also defines restrictions in the system for its execution, which allows minimizing the cost of the distribution. Such as: estimating the time of entry and exit of a vehicle, extra time, differences of the vehicles capacities, better execution of the routes to decrease to the smallest number of vehicles, route's time and traffic, calculate the time of delivery per customer identifying difficulties in discharge, integrate the delivery sequence into a WMS in order to perform a order in loads, and monitor driver's consecutive work hours.

2.5.1 *Company TMS benefits*

All of the features listed above are benefits related to TMS, and yet we are always able to add more features to user requirements. In the following text are presented some TMS benefits and impacts:

- Reduce transport costs and improve service level.
- Improve transport resources usage.
- Improvement on charge and routes planning.
- Less time to plan distribution and charges positioning.
- Availability to estimate a better route cost due to previous samples.
- Availability of online information.
- Bills payment security and transport security.
- Real-time costs knowledge, helping on business financial results.

The TMS benefits are defined by the necessity of the company when they solicit better management in the processes. The transportation service is a crucial point in the distribution directly related to the customer satisfaction index.

Once companies finish the delivery, they cannot have feedback with satisfactory performance of their internal and external processes. TMS tries to offer the easiest way to manage the components as a supplier. The company must understand IT importance to avoid incorrect evaluations.

2.6 AUTOMATIC VEHICLE LOCATION

AVL is a system that defines where the GPS technology is used to vehicle monitoring, which consists of GPS receiver and emitter, and comes from an extensive class of system called ITS.

These systems are the result of the assembly of technology and equipment that allows automatic and centralized determination, viewing, and controlling the position of multiple vehicles. AVL systems have been used for more than two decades, from the location of emergency vehicles to delivery vehicles.

According to Hermida (2012) AVL works as it follows: The information collected is transmitted to a control center, which is usually located at the organization's headquarters. Being the data transmitted between the vehicle and the base using conventional mobile GSM or satellite networks.

Generally, the GSM system is the most economical and suitable for data communications, with the highlight being the distribution of goods and new counters of *Energias de Portugal (EDP)*. In addition to the position monitoring, it is also possible to send information to the vehicle to control the lighting, fuel, or even a panic button if something goes wrong by immobilizing the vehicle. A route eventually established may be a factor in dispelling a state of emergency by the detection of a wrong route or high speed.

One of the crucial factors of AVL is precision, operating in absolute mode, with an error of around 50 meters. However, today, this precision is only influenced by buildings, interference, or tunnels. In order to increase precision, it is necessary to use *Global Positioning System Differential (DGPS)* in real-time, using multiple base stations, with this method, the accuracy stabilizes around 2 meters. Another way is *Dead Reckoning (DR)*, which works on direction sensor with the vehicle odometer, operating when the GPS signal is interrupted, the positions are calculated according to the last position of the azimuth obtained by the direction of the sensor and the distance traveled by the vehicle odometer. Thus, with the combination of DGPS and DR, we can increase accuracy and ensure a position in any environment.

2.7 GLOBAL POSITIONING SYSTEM

GPS is a satellite-based radio navigation system provided by the United States Department of Defence. It provides unmatched accuracy and flexibility in positioning for navigation. This system uses a constellation of 24 satellites to give a user an accurate position on earth. A GPS receiver (user) on earth will receive signals from these GPS satellites and from the received signals are able to calculate its position on earth.Gopi (2005)

Its application is obviously in general and commercial aviation and maritime navigation having the benefit find the way to the place or know the velocity of displacement. This system is very present on vehicles with map navigation, making possible a vision area.

It is essential to define "accuracy" due to a soldier in a desert accuracy means 15 meters, and to a ship in coastal waters, accuracy means 5 meters. Nevertheless, to a land surveyor, accuracy means 1 cm or less. GPS can be used to achieve all these accuracies in all these applications by employing a suitable type of receivers and relevant techniques.

2.8 BLOCKCHAIN

2.8.1 *Introduction*

Blockchain is one of the greatest innovations of the current century, being adopted to the various relevant sectors of the industry. Its implementation goes back to the early 1990s, when Stuart Haber and W. Scott Stornetta began working on a cryptographically secured chain of blocks, where no one could tamper records of the date and time of documents.Haber and Sornetta (1991) The update for the Merkle trees was in 1992, allowing to collect more documents in a single block. Over the years blockchain begins to gain popularity in 2008 when Satoshi Nakamoto released a whitepaper about the technology, that provided details about the increment of digital trust and given the decentralization aspect that meant no one would ever be in control of anything.Bashir (2018)

2.8.2 *Definition*

Blockchain is a distributed database of records and is the technology behind Bitcoin, which can be defined as a data structure that operates transactional records, ensuring safety, transparency, and decentralization. The transactions in the public ledger are verified by the consensus of a majority of the participants in the system.Crosby et al. (2015) Moreover, once entered, information can never be erased. The transactions on the Blockchain are secured by a digital signature that verifies its authenticity.

All network participants on the Blockchain are allowed to reach an agreement, usually known as consensus, and the information recorded is available for all the network participants. The protocol of Blockchain operates on top of the Internet, is a *Peer-to-peer (P2P)* Network of machines running a protocol-based, maintaining an identical copy of the ledger of transactions, enabling P2P value transactions without an intermediary through a machine consensusVoshmgir et al. (2017).

2.8.3 *Types of Blockchain*

The Blockchain can be divided into three categories: Public, Private, and Hybrid, as it can be observed in figure 3 that represents the types of Blockchain that are most appropriated for diversified areas.

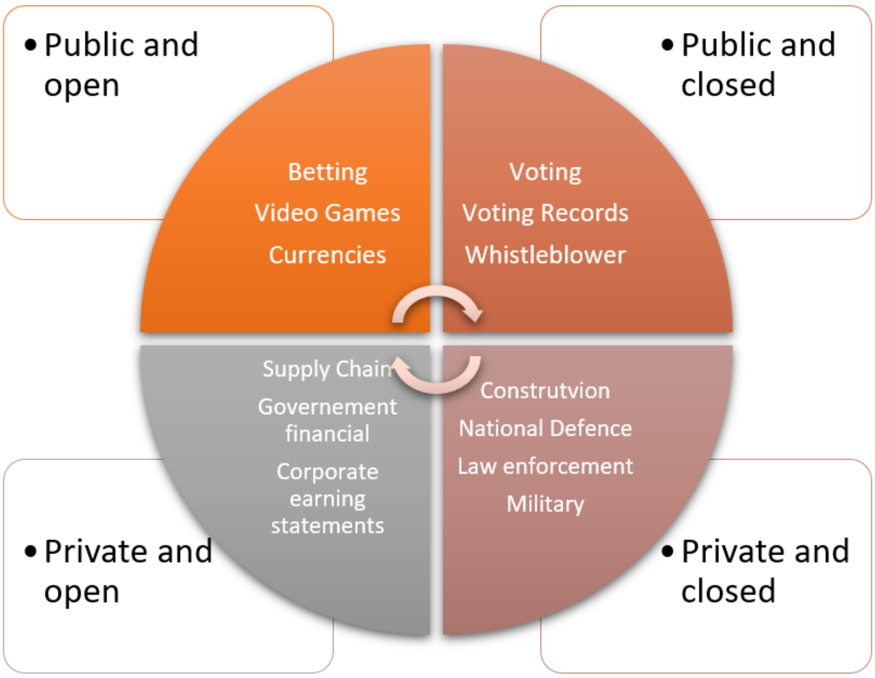

Figure 3: Blockchain types and appropriate areas. Adapted from: Massessi (2019)

Public blockchain or Permissionless blockchain

The information is opened to the public, and anyone can participate, without permission, the protocols are based on *Proof-of-Work (POW)*, and no one has control over the network.

Everyone has the full right to download the code and run any public node on their device. The current status and add blocks to the chain can be performed by anyone, as the transactions are transparent and anonymous. The system or servers do not need maintenance. That is the reason why decentralized applications do not cost much to build and run. Voshmgir et al. (2017)

Examples: Bitcoin, Monero,Etherum,Litecoin.

Private blockchains or Permissioned blockchain

The information is not accessible to the public because the peers do not have access to the blockchain network, whereas only a member of the group can read and send transactions. For example, the participation of a node is defined by that group, extending that groups and participants can verify transactions internally. One of the advantages of a private blockchain is entering sensitive or business-critical information since the information cannot be exposed to anyone.

It is a centralized system, but sometimes that idea in most cases is limited to the number of nodes in the network, which contradicts the idea of a centralized system. With private blockchains, the risk of security breach increases, and the cost of the transaction is lower. Voshmgir et al. (2017)

Examples: Multichain, Blockstack, Hyperledger, Corda, Exonum.

Consortium or Hybrid Blockchains

The combination of public and private is usually designated as a hybrid, it can only be accessed by a group of individuals or organizations that have decided to share information with each other.

Consortium control multiples aspects of the blockchain, such as the validation of transactions, managing node privileges, smart contracts, an extension of nodes, affirming that the entire network is defined and governed by the members and nodes. Leonard (2017)

In the table below, the blockchain attributes types are referenced more precisely, such as of best efficiency and which is adequate for several purposes.

	Public	Private	Consortium
Structure	Descentralized	Centralized	Partially Descentralized
Access	Open Read/Write	Permissioned	Permissioned
Speed	Slower	Faster	Depends number of nodes
Consensus	Proof of Work or Proof of Stake	Pre-approved	Pre-approved
Identity	Anonymous	Known Identities	Known Identities
Use Cases	Cryptoeconomy	Reference Data Management	Secure data sharing

Table 2: Types of blockchains

2.9 CONSENSUS

As mentioned above in section 2.8.2, Blockchain is a distributed decentralized network, providing security, transparency, and decentralization unified over a P2P network. The consensus is a general agreement among all participants in a group, in which everyone accepts and supports decisions.

In Blockchain does not exist a central authority to validate and approve each transaction, identifying this failure, the need emerged to implement a consensus mechanism which is a fundamental part of any Blockchain Network.

According to Bashir (2018) consensus mechanism is a protocol that ensures which information on nodes is synchronized with each other. All peers on the network agree about the present state on the distributed ledger alternatively of one master computer (centralized). Therefore that reliability is increased, existing a correlation with reliable peers in distributed computing.

In brief, the consensus mechanism ensures that a new block inserted into the Blockchain is unique and there is only one version that is accepted by all nodes on the Blockchain network.

There are some aspects of security and issues that can be supervised or expected such as, failures of nodes, message delays, corrupted messages, network split, and malicious nodes, being that the probability of an attack is higher if the consensus mechanism is not suitable.

2.9.1 *Types of consensus*

Achieving reliability on distributed computing implies an enigma due to the presence of faults or corrupted data during the consensus process, many mechanisms of consensus are created in order to adequate the most suitable mechanism to the system in question.

Some consensus algorithms will be referenced during this dissertation, including the principal and suitable following this subject, in order to resolve the mentioned problems on section 2.9.

Proof-of-work (Bitcoin)

Used in cryptocurrencies, more specificaly on Bitcoin and Ethereum, is the first and common popular mechanism adopted to mine cryptocurrencies. The process is reached by a group of miners solving a mathematical problem determining how peer does the work according to the quantity of computer power (hash rate), allocating a percentage to be fair.Bashir (2018)

The probability of getting the ”task” is according to the higher hash rate on the pool (a network of miners). After that, a new block is created, and the transaction is approved.

Some limitations may occur depending on the type of machine. To have a notion, a simple machine, can do approximately between 20 and 50 transactions per the second, which corresponds to an indefinite time to generate a block.

Proof-of-stake (Etherum)

Proof of Stake (POS) appears to substitute the significant problem of POW, such as the higher computational consumption on mining operations.

A deterministic way creates the new block, the validator (someone who can produce blocks) is selected pseudo-randomly to create a block. In this system blocks are usually designed to be forged, unlike pow that is mined.

To validate a block, the cryptocurrencies will be detained in the networks as proof of their participation(stake).

One user which contains more stakes corresponds to a new probabilistic way to choose a forger. When a node is chosen to produce a new block, it shall be responsible for checking that transactions are valid on this block. In the absence of any irregularity, the block is added to the Blockchain. Bashir (2018)

Proof-of-elapsed-time (Intel Sawtooth)

PoET is an algorithm used mostly on a permissioned blockchain network, that means every node in the system needs to get an authorization for accessing the network.

The main principle of this algorithm is the identification of node, which influences the mining rights, for example, to choose who becomes to produce a block, each node is allocated with a random waiting time, where the node with the longest waiting time will be the first to be chosen on the ledger to produce a new block.

Hyperledger sawtooth uses this consensus mechanism with the proposal for further development.

Byzantine Fault Tolerance and variants – Hyperledger Fabric.

The Byzantine history goes back many years ago, when generals circled a city for an upcoming attack, ignoring the coordination.

This problem references a trusted consensus by confirming that the information referenced is valid and trusted. If a community member sends unreliable information, the trustily of blockchain is broken, not being possible to correct faults because there is no central authority.

The main objective of *Byzantine Fault Tolerance (BFT)* is finding a solution to prevent a weak or unreliable node. With this methodology, the possibility of reaching a consensus when nodes respond with manipulated information or not respond is probable.

Practical Byzantine Fault Tolerance

The PBFT algorithm projected by Miguel Castro and Barbara Liskov is developed for asynchronous consensus systems and is improved in general to solve all problems on Byzantine failures. The nodes are sequentially ordered, such as a leader node (primary node) and backup nodes (secondary), each node can become primary if the secondary nodes fail. Both network nodes can communicate with each other with this order. Castro and Liskov (1999)

Figure 4 evidence how the algorithm works, first the client sends a request to the leader node, and this node, broadcasts the request to the backup nodes, after that the nodes (leader and backups) execute the request and send back a response to the client. The solicitation is attended successfully when the client receives "m+1" replies from diverse nodes in the network with the same result. The m represents the maximum broken nodes permitted. Hooda (2019a)

Figure 4 shows the four phases of the algorithm.

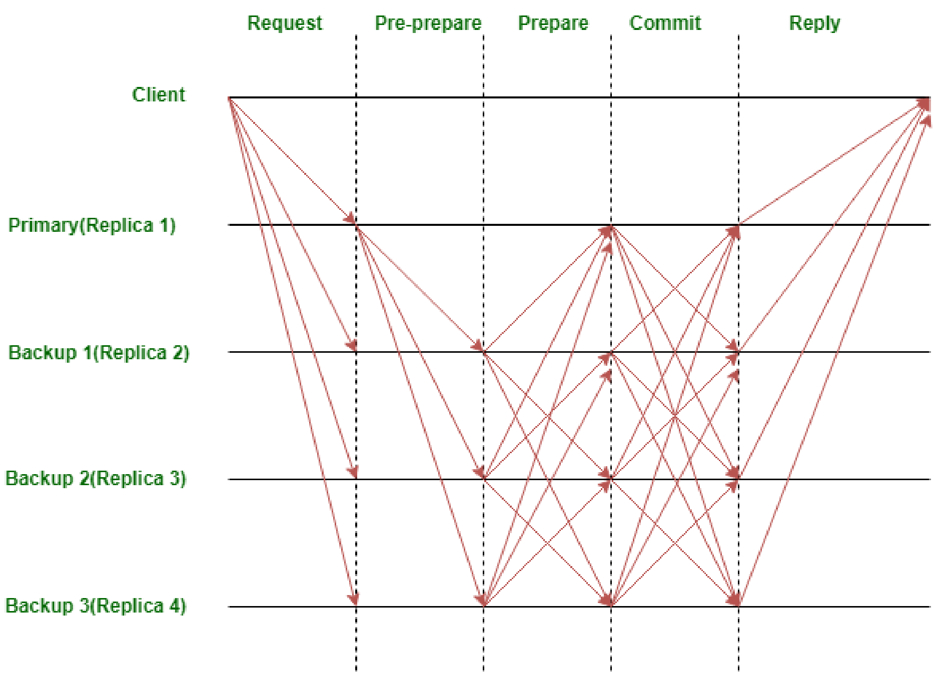

Figure 4: Steps of PBFT Source: Hooda (2019a)

Therefore, this algorithm is best suited for permissioned blockchains as the network is not very scalable, and communication with the nodes is always guaranteed. Some advantages and disadvantages are mentioned below.

Advantages:

- Reduction of energy- no mathematical problem is being solved (hash)

- No need for confirmation - The transaction can be finished if the nodes approve the specific block. The consensus is reached because genuine nodes can communicate with each other at the same time. Anwar (2019)

Disadvantages:

- Sybil attacks – The probability of network attack is reduced if the number of network nodes increases.
- Communication gap – The nodes can send many messages during a request, and if the number of nodes increases, the probability of losing a message is higher.
- Scalability - It only works efficiently for a smaller group of nodes.

Kafka

Apache Kafka Data is an open-source application that focuses on a distribution platform, such as a message queue or a corporate messaging system. Originally it was developed to create a system with low latency and a high transfer rate. It was implemented on Hyperledger Fabric, and it is used as the consensus mechanism, which consists of nodes agreeing with the same order of transactions, and providing crash fault tolerance to ordering service. It is like Raft because only the leader does the ordering, and if a leader node crashes, the system needs to elect a new leader.

The process flow consists of a multiple order that must communicate with the Kafka cluster, which ensures that the ordered process receives transactions and create blocks in the same order. Hyperledger (2019a)

Kafka was available since Fabric V1.0 and is no longer used as primary on a new version of Hyperledger V1.4.2. It is not recommended for Production because it does not provide decentralization.

Raft

The Raft is a consensus algorithm that is developed to supervise a replicated log, is designed to be easily understood. It is a *Crash Fault Tolerance (CFT)* consensus, which assumes the leader always act correctly.

The client requests are sent by leader node to the followers directly, and if a majority of followers confirm replication, the request is committed, and also apply to their local state machines. Hooda (2019b)

The process of leader election is decided at the start with randomized timers. All nodes start in the follower state, accepting and receiving logs from the leader (if already got elected). The follower node becomes the Candidate State when it does not have any message from the Leader node. For a candidate to be elected, a minimum vote is necessary from other nodes. Normally the leader node periodically sends a message to its followers to maintain authority. Hyperledger (2019a)

2.9.2 *Review and comparison of consensus types*

Since there is a range of different consensus algorithms, it is decided to choose the most common algorithms used nowadays from different technologies of blockchain. Table 3 shows and cites an overview of the best characteristics of the consensus algorithm.

Consensus	PoW	PoS	PoeT	Bizantine and variants	Kafka	Raft
Examples	Bitcoin	Etherum	Hyperledger Sawtooth	Hyperledger Fabric	Hyperledger Fabric	Quorum, Hyperledger Fabric
Type	Permissionless	Both	Both	Permissioned	Permissioned	Permissioned
Node identity	No	No	Yes	Yes	Yes	Yes
Energy Saving	No	Yes	Yes	Yes	Yes	Yes
Cost	Yes	Yes	No	No	No	No
Scability	High	High	High	Low	Low	Low
Trust model	Untrusted	Untrusted	Untrusted	Semi-trusted	Trusted	Trusted
Transaction	Probabilistic	Probabilistic	Probabilistic	Immediate	Immediate	Immediate
Fast Consensus	No	No	No	Yes	Yes	Yes
Transaction Approval	Long	Fast	Fast	Fast	Fast	Fast
Verification Speed	>100s	<100s	<100s	<10s	<10s	<10s

Table 3: Types of consensus mechanisms

2.10 SMART CONTRACTS

A smart contract is an application running on top of a blockchain with a preprogrammed set of instructions, supporting the negotiation or performance on a transaction. The terms of the agreement of contract between two or more parties are executed automatically when certain conditions have been met.

It is a possibility of using the benefits of the blockchain's computational efficiency and security to program systems that today work using other technologies. When Bitcoin was launched, the advantages of blockchain technology were revealed to the world, but it was explicitly in financial transactions. The smart contracts expanded this to other applications.

This technology can be used to register land purchases and even send money from a source to a destination, without the necessity for trusted intermediaries formalizing the relationship between people and institutions. Voshmgir (2019)

Image 5 represents a typical scenario when a Manufacture needs to send an item to an end customer through a transport company. A smart contract automates the process between the companies because they have stored and replicated documents during the transaction, ensuring authenticity.

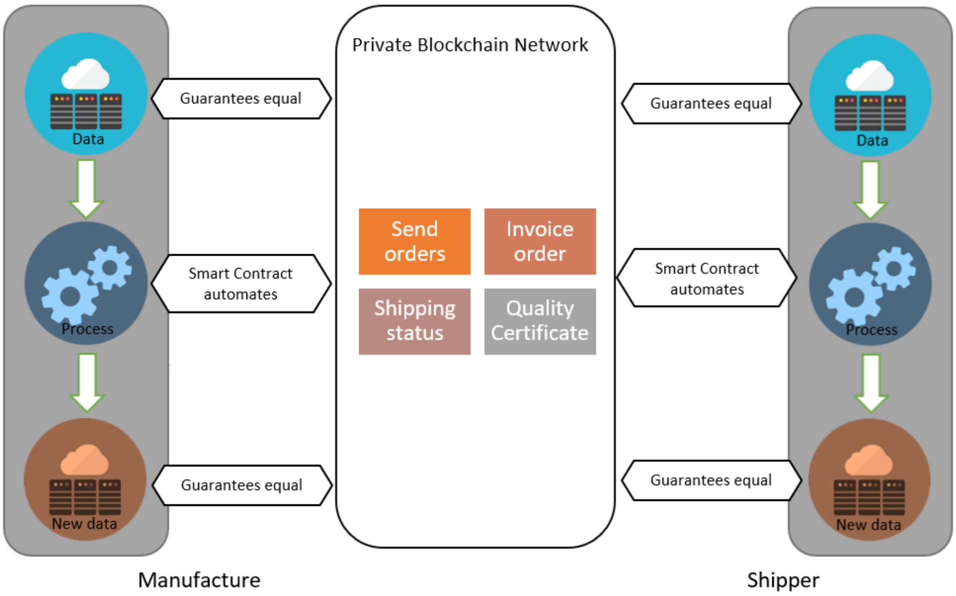

Figure 5: Blockchain model with smart contracts, Adapted from: Itweb (2019)

2.11 BLOCKCHAIN PLATFORMS, AN OVERVIEW

2.11.1 *Ethereum*

Ethereum is a platform that was founded/presented by Vitalik Buterin, through a white paper, in January 2014, although it was only released on July 30, 2015 Buterin (2014). According to Vitalik's white paper, the objective of Ethereum is to improve and merge concepts like altcoins, scripting and on-chain meta-protocols, which promotes to create "arbitrary consensus-based applications that have the scalability, standardization, feature-completeness, ease of development and interoperability offered by these different paradigms all at the same time" Buterin (2014).

This decentralized platform is capable of running smart contracts and decentralized applications (Dapps) using blockchain technology. Smart contracts, similar to cryptographic ″boxes″ contains values that can be unlocked if certain conditions are established, providing more power than that offered by Bitcoin scripting. ″Ether″ is the primary internal crypto-fuel of Ethereum. In this way every transaction, record, execution of distributed code, the signature of a digital contract, or any other application that runs on the Ethereum network is paid in ether. So Ethereum can be considered a giant computer ('World Computer') in which users pay for the amount of resource used.

Dapps are applications that work correctly as programmed without interference from third intermediates in any existing centralized service by allowing anyone to rely on an unknown counterparty to do 'Agreements' in a 100% digital way.

In addition, one crucial point that differentiates Ethereum from the Bitcoin (the most widespread digital currency) and makes it different from the altcoins, in general, is the aim of taking the blockchain technology and the smart contracts to ″all″ that can be programmed.

What is the functionality Ethereum?

One of the purposes of Ethereum being created was to be a smart contracts platform, thus allowing to overcome one of the limitations of Bitcoin (a very limited smart contract platform). Smart contracts:

- They function as "multi-signature" accounts, so funds are spent only when a required percentage of people agree.
- Manage agreements between users.
- Provide utility for other contracts (similar to how a software library works).
- Store information about an application, such as domain registration information or membership record.

One of the contracts that run on *Etherum Virtual Machine (EVM)*, the formal definition, was written by Gavin Wood. The EVM offers a ″Turing Complete″ programming language that means it supports more comprehensive programming than the Bitcoin network and can run scripts using an international network of public nodes. An expression used for the EVM is a distributed global computer where all smart contracts are executed.

Differences between Ethereum and Bitcoin?

Ethereum and Bitcoin are both a distributed public blockchain network. However, there are some significant technical differences. In particular, they differ substantially in purpose and capability. At the same time, Bitcoin provides a particular application of blockchain, a point-of-point ATM system that allows online payments from Bitcoin to Ethereum blockchain to execute the programming code of any decentralized application.

In the Ethereum blockchain, instead of bitcoin miners, miners use this technology to gain Ether, a type of cryptographic token that feeds a network. In addition to negotiable encryption, Ether is also used by application developers to pay transaction fees and services on the Ethereum network. There is a second type of token that is used to pay mining fees for including taxes on your block, called gas.

Framework	Ethereum	Bitcoin
Concept	Intelligent contracts	Digital Money
Transaction	Send Alex to Daniel with condition: run on 1 jan 2019 if balance $<$ 10 eth	Send Alex to Daniel
Founder	Vitalik Buterin	Satochi Nakamoto
Released on	July 2015	January 2009

Table 4: Ethereum vs Bitcoin

2.11.2 *Openchain*

Openchain technology is embedded in one of the various branches of Blockchain. However, Openchain does not use the concept of a block data structure as it is used in Blockchain, but rather by a direct linking of transactions no longer grouped in blocks.

There are several Openchain's characteristics, which also make it an added value, namely: the instant confirmation of transactions, also transparency and auditability of them with no mining fees. It also allows for excellent security through the use of digital signatures. It presents multiple levels of control, and the Ledger can be open or closed or even a mixture of both such as :

- An open ledger that can be anonymously united.
- Closed-loop Ledger which means only an administrator can approve a participant.
- Presence of an open and closed Ledger there are anonymous users and approved users, the approved ones enjoy more rights.

This technology also presents high scalability and capacity to have multiple instances of Openchain, where they are replicating each other. Besides that, Openchain uses a hierarchy of accounts, which adds many management options that other systems do not present like Bitcoin.

In summary, Openchain technology is based on distributed open-source ledger technology being an added value for those who want, a secure and robust, to issue and manage digital assets. Charlon (2017)

2.11.3 *Quorum*

Quorum consists of a private/permissioned blockchain, which is based on the official Go implementation of Ethereum protocol Go-Ethereum WebSite. However, one of the main objectives of the Quorum design is to make minimal changes to the go-ethereum, so that they remain in sync with the innovations that are occurring from the public Ethereum codebase.

This technology uses a consensus algorithm, which is 'raft-based' that promotes block creation, faster block times, and transaction finality. Quorum intends to use encryption so that only those who participate in the transaction are enabled to see sensitive data. For this, the basic idea combines the involvement of a single shared blocker and an intelligent contract software architecture (provides private data segmentation) and modifications to the go-Ethereum code base (the least possible because of the reasons already mentioned) such as modifications to the block proposal and validation processes. The change of the block validation process allows the validation of public transactions and any private transactions (those to which they are party by executing the contract code associated with the operations), through all nodes. In the case of other "private transactions", a node will ignore the process of executing the contract code.

Thus, it will be obtained a segmentation of the state database that is divided into a private and a public state database. In addition, the transactions contained in the blockchain are replicated in their entirety (across all nodes) and protected cryptographically (for immutability). However, the client node's state database no longer stores the state of the entire global state database.Morgan (2018)

2.11.4 *Ripple*

Ripple is a payment system decentralized, which is based on credit networks. Anyone can implement this system since Ripple code is an open source and available to the public.

In the Ripple system, nodes can assume up to three functions:

- An open ledger that can be anonymously united.
- Market makers.
- Validate servers that run the Ripple consensus (check and verify all transactions that occur in the system).

A more specific circumstance, if a user wants to send a payment to another by money transfer in XRP (Ripple's currency, cryptographically) can also do so using any other currency. However, for the latter type of payment, made in non-XRP currencies, Ripple only records the amounts due between the two entities, called distributed credit network system. Important note, Ripple users are usually referenced through pseudonyms and have a public/private key pair.

In the Ripple system, there is a control of all transactions made in the system through a distributed ledger. Armknecht et al. (2015)

The Ripple ledgers present the following information:

- The set of transactions.
- Information related to the account, like settings, total balance, trust relation.
- The timestamp.
- Validation of the ledger (by a status bit indicating whether the ledger is validated or not).
- A ledger number.

About the validation of the ledger, the ledger is referred to as open if it is not validated yet. However, the most recent validated ledger is considered as the last closed ledger.

Ripple type of transactions

Payment, which consists in sending money from one account to another, is considered the most common type of transaction. Besides that one, it will be considered the other five types:

- AccountSet: allows an entity to define relevant options for its account.
- SetRegularKey: enable an entity to change/set the key used by the entity to sign future transactions.
- OfferCreate and OfferCancel: expresses an intent to exchange currencies.
- OfferCancel: removes an offer from the ledger.
- TrustSet: creates/or modify a trust link between two accounts.

Consensus protocol

Ripple has a consensus protocol that is considered asynchronous and round-based, which is executed by the network validation servers. The consensus protocol comprises three phases: the collection phase, the consensus phase, and the closing phase of the ledger.

Ripple vs. Bitcoin

In this part, it will be presented a brief comparison between Ripple and the well-investigated Bitcoin system about two essential parameters: the security and privacy provisions.

Security: Ripple to ensure the authenticity of system transactions also uses (like Bitcoin) *Elliptic Curve Digital Signature Algorithm (ECDSA)* signatures. As already mentioned, and too similar to Bitcoin, Ripple has all the transactions and respective orders to the public access, because it is an open payment system. The log of all transactions are verified by the validation servers that vote for the correct transactions in the system (one vote for each validation server), so transactions where the validators agree 80% are considered valid . However, according to Ripple Labs, it is easy to identify colluding validators and in this case, there is no formal security treatment for the correction of the Ripple consensus protocol. Armknecht et al. (2015)

Bitcoin safety has been thoroughly investigated in several studies and is better understood than Ripple. In Bitcoin, transaction security is guaranteed through POW, which replaces the vote by validating the notion of Ripple on the server. Also, unlike Ripple, if you confirm transactions in the global ledger (i.e. transactions receive six confirmation blocks), it is computationally impossible to modify those transactions.

Privacy and Anonymity: In this case, both open systems, Ripple and Bitcoin, are instances of payment systems. Here, the anonymity of the user is guaranteed, and users are also expected to have several accounts (corresponding to different pseudonyms).

However, we must keep in mind that in the case of Ripple, payments usually have only one account as input, unlike Bitcoin, which has several. Although there is a protection of the users' identities, the transactional behavior of the users (time and amount of transactions) is leaked in the process and the transactions are publicly announced in the system. Thus, it demonstrates the privacy limits in open payment systems. There are also several proposals to improve the privacy of users in these systems, with the majority of the projects using zero-knowledge proofs and cryptographic accumulators, and thus promoting the avoidance of tracking network expenses. Although most of these studies focus on the Bitcoin system, it is argued that they apply equally to Ripple.

Clients, Protocol Update, and Maintenance: Ripple and Bitcoin allow any entity to build and release their software client to interface with any of the systems. However, so far as is known, there is no lightweight and secure version of Ripple. Still, Bitcoin clients also can view on devices with limited resources, such as mobile phones, allowing simple payment verification of Bitcoin.

2.11.5 *Hyperledger*

Founded in 2015 by Linux Corporation the Hyperledger is an open-source project created for the propose of fulfilling requirements of the industry, the lack of support for private transactions and the scalability of public blockchain was one of the first requirements for this development. The creation of modular tools allows developers to implement software based on a distributed ledger. Many companies like Intel, Airbus, Samsung, IBM are contributing to Hyperledger from the fields of *Internet-of-things (IOT)*, SCM, manufacturing, and production.

The smart contracts in Hyperledger are used to create tokens, similar to cryptocurrencies, which are used on permissioned blockchain.

All of us connected to the Hyperledger network imagine the applications can be modular, open-source platforms that are easy to use. According to IBM, it is possible, and we aim to create an environment that enables their imagination a reality. IBM (2018).

In this section, types of Hyperledger frameworks will be discussed that are used to build an optimized logistics application, which are the best framework option to adopt on this dissertation.

Frameworks

Hyperledger promotes a variety of technologies with these fundamental requirements: private transactions, interoperability, portability, and identity.

The various projects mentioned below include a variety of features that can be used in various branches of industry.

Hyperledger Burrow

Currently, in development, the main feature of Burrow is the fast transaction throughput on a proof-of-stake consensus engine. In this case operates with EVM, which allows a modular blockchain client developed with Monax, the smart contract code is executed explicitly on a permissioned virtual machine with Ethereum base. Hyperledger (2017)

Hyperledger Fabric

Fabric is a permissioned blockchain building for distributed ledger solutions created by IBM and Digital Asset with the purpose of producing an easy-to-adopt platform for large industrial networks. The new blockchain architecture aims at resiliency, flexibility, scalability, and confidentiality.IBM (2018)

Providing a modular architecture, the execution of distributed applications is possible in different nodes on a network like a distributed operating system. Many smart-contracts, which are called chaincode on Fabric are written in standard programming languages like Java, JavaScript, or golang, allowing interoperability in the different nodes.

The most significant advantage in Fabric is confidential transactions by different entities without going through a central authority. This confidential transaction is operated by channels, which enable a group of participants to create a separate ledger of transactions. Hyperledger (2017).

Hyperledger Indy

Created in a business scenario, Indy is a blockchain-based system that is solely focused on digital identity on the internet, which proves in a real scenario a genuine identity with specifications and design patterns for decentralized ledgers.Hyperledger (2017)

Hyperledger Iroha

Iroha is a blockchain framework used for easy integration in enterprise and incorporation into infrastructure projects, and the distributed ledger offers a development environment in C++ that can be used for small data and high-performance data. One of the benefits of Iroha is the incorporation of Fabric, Sawtooth, and many other projects, which provide features that may be useful to final consumers. The storage level is composed of three different components: Redis (block index), Flat files (block store), and PostgreSQL (world state view).

Hyperledger Sawtooth

The Hyperledger Sawtooth is an adaptive platform designed for high scalability and adaptability running in distributed ledgers, including a consensus algorithm named Proof of Elapsed Time. With this, the state of the transaction-based updates can be shared between untrusted parties. The distributed ledgers allow the possibility of digital records without a central authority.

Actually, typical blockchain-based systems use a single platform to execute many applications and cores, which may result in performance problems and security issues. With Sawtooth, the core ledger is separated from the application environment, increasing security, and a possibility to develop any application. Hyperledger (2017)

2.11.6 *Hyperledger Fabric*

Typical scenarios on Hyperledger offer a modular architecture which is adjusted to the user or organization's requirements. An example of this are the permissions of involved entities, smart-contracts called chaincode, and consensus mechanism referred to as ordering service on fabric.

A fundamental part of *Hyperledger Fabric (HF)* are channels, resulting from the creation of separate ledgers of transactions, promoting communication between organizations on the same channel, this component will be later referenced and detailed.

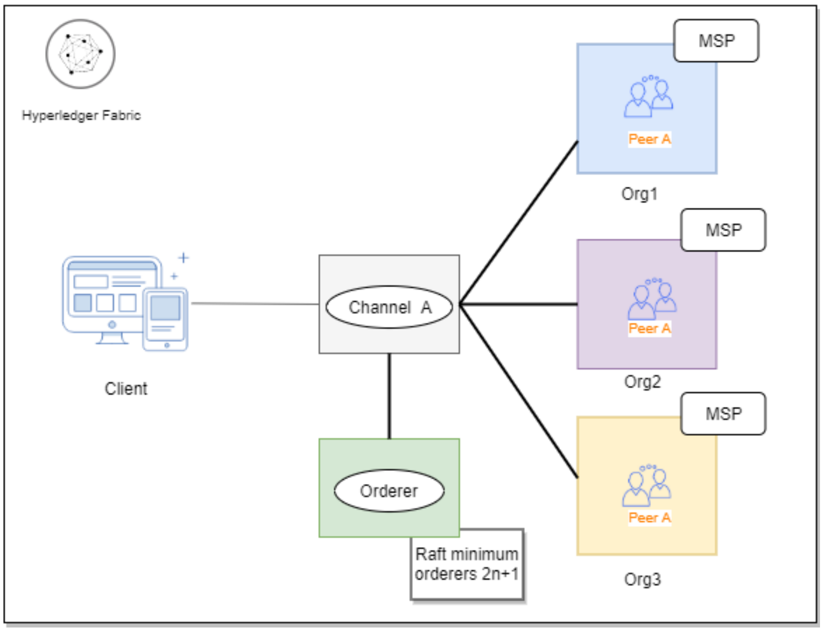

Figure 6: Fabric Architecture, Adapted from: Thummavet (2018)

In figure 6, a simple architecture on Fabric will be represented, composed by three organizations on the same channel and each organization contains one Peer, an orderer, client, and a *Membership Service Provider (MSP)*. The major components will be explained with a sequential order.

An organization can be a hospital, manufacturer or shipper, and it depends on the context that it provides, each organization comes with pre-defined key *Public Key Interface (PKI)* and a *Certificate Authority (CA)* that it is necessary to authenticate users in organizations, one of the best practices is using a MSP for each organization. A peer in an organization can contain a copy of the chaincode, execute it, and maintains the state. There are multiple types of peers with a different role that will be specified in the next section.

The client is an application, which communicates via API REST to the blockchain network, with the task of invoking transactions and presenting information in JSON format. To ensure authentication and network access, the MSP manages user IDs and allows a specific user to access the blockchain network.

Peers

As mentioned in section 2.11.6, different rules are attributed to multiple peer nodes. A peer can be considered a node, and their main functions are receiving transaction proposal requests and maintains a copy of the ledger.

In this section the different types of peers and their rules will be referenced.

Anchor Peer

Each member on an organization in Fabric Network has an anchor peer (or multiple anchor peers to prevent a single point of failure)Elrom (2019), that are configured on the initial configuration. With anchor peer, the network was able to discover through the gossip protocol enabled participants on the channel and communicate with other peers across different organizations.

Committing Peer

These peers are responsible for committing the block from Ordering Service and keeping the current state of the ledger. They do not have a chaincode installed so they cannot run smart contracts or invoke chaincode. The block from ordering service contains a list of transactions that are validated by committing peers. The *Read and Write (RW)* sets are checked and verified if they correspond to the current world state.V (2019)

Endorsing Peer

Endorser peer is responsible for endorsing a transaction request from a client and simulating then. When receiving requests to validate each transaction endorser peer prepares a transaction proposal and sends it to the orderer peer (Consensus).Elrom (2019) All endorser's peers contained a copy of a chaincode (smart-contract) and needed that installed for the proper functioning of the network.

Leader Peer

A leader peer is responsible for receiving new blocks from Ordering Service to minimize overload on the Orderer. Hyperledger (2019a) Peers in the same organization can receive a block from leader peer using gossip data dissemination and check communications to make sure that peers connected got the message.

There are two types of leader peers, static and dynamic, that are defined in the initial network configuration. Static is fixed, and if they fail a new election does not exist, the dynamic one is used if a peer is not available on the network. When this occurs, a leading peer is randomly selected or elected by voting.

Autorization (MSP and CA)

Authorizations in HF are regulated by MSP. The MSP is a service that determines which participants are validated and authenticated on the network. The MSP can manage single or multiple organizations and are identified by a MSP ID. The members on Fabric must know all identities in order to transact an operation because the network is private and based on permissions.Elrom (2019)

According V (2019) MSP uses a Certificate Authority, in order to verify and revoke user certificates upon confirmed identity as mentioned in figure 7. There are two types of MSP, a local and channel.The local one is responsible for managing users, peers, and orderers, with certain access rules. The Channel one defines administrative and user rights at the channel level.

As previously mentioned, CA is a module that was used for the MSP. Most organizations use Fabric's own certificate, and they can implement an external certificate in order to query or invoke a transaction. A CA is a component or module which is responsible for managing user certificates in order refusing users to prevent access to crucial information. These certificates use X.509 standard granting permissions and roles to users.Thummavet (2018)

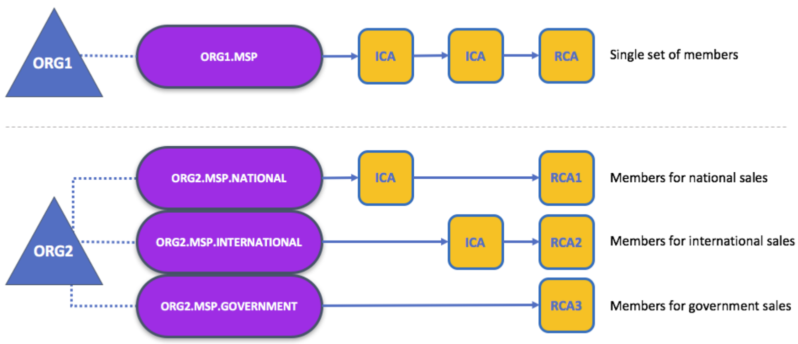

Figure 7: Membership service provider diagram, Source: Hyperledger (2019c)

Consensus

As it is mentioned in section 2.11.6, the HF Ordering Peer is responsible for Consensus on Blockchain. HF includes multiple types of ordering mechanisms so that the user can choose the desired ordering algorithm.

The consensus in this framework is different from the traditional consensus, which supports an exclusive endorsement model. This model follows a methodology that transactions are executed on a set of peers and follows an Execute-Order-Commit model Liew (2019). In HF, the consensus is based on these three specific steps:

Transaction endorsement, Ordering, Validations and Commitment.

These three steps are fundamental to the transaction life cycle. The first step endorses transactions through transaction simulation while the second one receives the simulated endorsed transaction and chooses the sequence of which transactions are written in the ledger, finally on last step committing peers are responsible for the transaction validation check from the orderers, after this validation committing peers are able to commit transaction to the ledger.Bashir (2018)

Figure 8 is an example that details the Fabric architecture transaction flow steps.

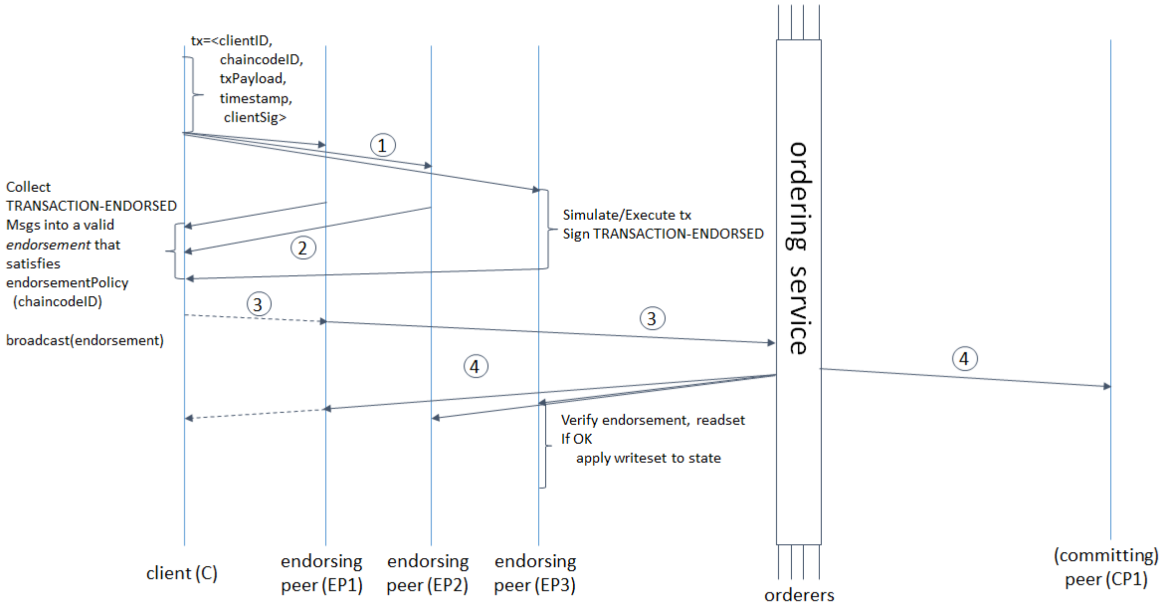

Figure 8: Fabric transaction flow (common case), Source: Margheri and V.Sassone (2018)

1 - A client (Application) sends a transaction proposal to Endorsing Peers, and each endorsing peer simulates a proposed transaction, without updating the ledger.

2 - Endorsing peers can capture Read Written data, called RW Sets. These RW Sets capture what has been read from the current world state and attach what will be written if the transaction was executed. .

The transaction Endorsement that was mentioned in the section above refers to a signed response to the results of the simulated transaction. The transaction method depends on the endorsement policy, which is specified on chaincode, for example:

a. AND ('Org1.member', 'Org2.member') – both organizations must sign to accept transaction

b. OR ('Org1.member', 'Org2.member') – one organization is demanded to accept a transaction.

c. OR('Org1.member', AND('Org2.member', 'Org3.member')) - require both signatures from Org2 and Org3 or one signature from an Org1 member. Hyperledger (2019a)

After the organizations accept the transactions the RW Sets shall be signed by endorsing peer and returned to Client (Application) to be used in the future.Liew (2019)

3 - After the Client has submitted the endorsed transaction, RW Sets to the ordering service. An Ordering occurs across the network in parallel with endorsed transactions. Ordering Service takes endorsed transactions and RW sets and order this information into a block and deliver to all the committing peers.

Ordering Service does not process transactions but is responsible for accepting the endorsed transactions and defines the order in which these transactions will be committed to the ledger.

In the Fabric network, transactions must be written to the shared ledger in a valid order. The order of transactions must be established to ensure that updates to the world state shall be valid when they are committed to the network.

4- The committing peers approve the transaction by checking RW sets still match the current world state, a brief example, the read data that exist on simulation by endorsing peers corresponds to the current world state.

When a committing peer validates transactions, the ledger is written, and the world state is updated with the Write Data from RW Set.

Finally, the committing peers asynchronously report to the client about the status of the transaction, namely, success or failure. V (2019)

Privacy on data

Privacy on data is one of the essential questions when applied to critical information. Fabric is permissioned, meaning it is not accessible to anyone, and for supply chain management, data cannot be shared with all companies without consent. Blockchain performs transactions trusted and credible, making it easier by showing to companies how beneficial it is for the authenticity of the information.

With the continuous evolution of Fabric, some methods and technology behind have changed. The first method implemented was Channels, which consists of creating a separate ledger of transactions for a group of nodes, not being able to see transactions of members outside the channel providing privacy,Hyperledger (2019a).

Image 9 (referenced) above describes how channels work on Fabric.

In the image, the channel created and named "channel12" belongs only to the organization one and two, which means every transaction made in that channel is not visible for Organization three.

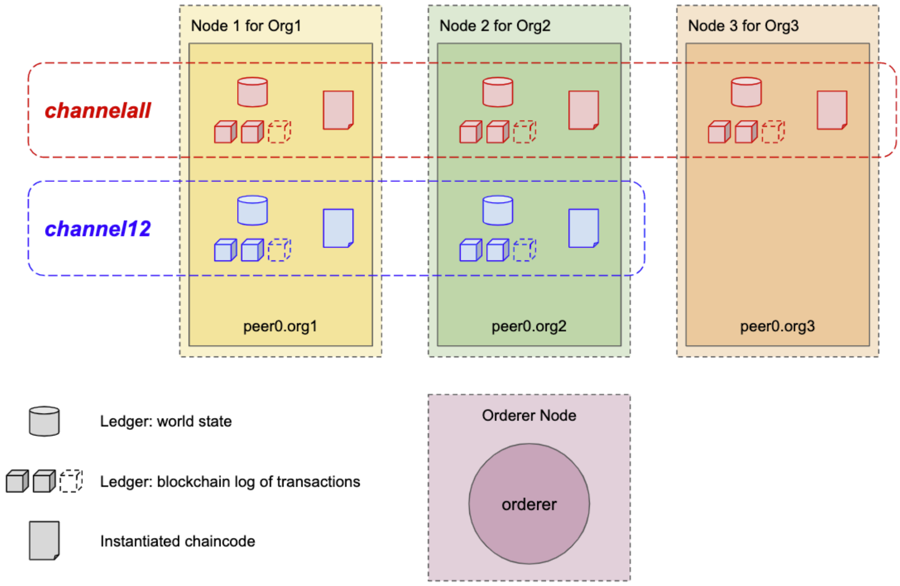

Figure 9: Channels example (common case), Source: Tam (2019)

Channel design principle will provide confidentially on a tunnel with a collection of peer nodes, as we can see in image 9.

Since version 1.2 of Hyperledger Fabric, a feature appeared called *Private Data Collections (PDC)*, providing data concealment, which means a peer in a node chooses which data can share between peers, and which peers are allowed to access the information. Nitin Gaur (2018).

The private data consists of information that we want to be kept secret from them and authorized organizations. One example is if we need to hide the price of products between organizations. Another example, enterprises can view trucks but cannot view what products the trucks contain.

Peer-to-peer communication using the Gossip Protocol is used for communications between the approved organizations, and data are stored in the peer's private database. According to Chaoqun Ma and Zhou (2019) et al. the private data processing is divided into two ways:

- First, ledger shall be kept confidential from off-channel members when the full transaction and new channels are required.

- Second, some organizations may have the ability to view some transactions, for authenticity purposes, not compromising the privacy of the data involved.

3

BLOCKCHAIN IN LOGISTICS

As mentioned in the previous chapter, Blockchain can be used for several sectors, from the financial area to logistics, depending on the types of requirements. Industry 4.0 consists of a new area that is in constant development, which includes many technologies that go with this dissertation by highlighting supply chain and IOT.

Blockchain is capable of being a viable solution for the development of Industry 4.0 due to the possibility to manage IOT devices in a viable way, on the logistical environment, highlighting, vehicle tracking, orders, and products. Edvar et al. (2019)

Hackius and Peterson studied the application of Blockchain on logistics. They did an online survey and asked logistics professionals for their opinion on the following use cases: barriers, facilitators, and the overall prospects of Blockchain in logistics and supply chain management.Hackius and Petersen (2017)

The questions in the following section are in agreement with specific restrictions that may appear during implementation.

3.1 QUESTIONS ABOUT IMPLEMENTATION

Where to start and how to implement Blockchain in logistics?

With the literature review and the direction of the dissertation, it will be decided to focus the methodology on road transport, not forgetting other stakeholders.

Following Dobrovnik et al. (2018), the adoption of blockchain is performed in several and different transformation stages, such as single-use, localization, replacement, and transformation, which need distinct levels of cooperation.

In order to make blockchain viable, in road transports, three main things may happen, according to Salama (2018).

- First, Blockchain must be used as a unique source of truth by everyone. This means every company or organization will be monitored and make secure contracts between parties involved allowing reliable data sources.

- Second, A large number of shippers and carriers should join in the blockchain, meaning these entities must be able to enter in blockchain, adopting new hardware and software. This focal point is essential because small entities do not have funds to adopt new technology.
- Third, Data standardization should be used by everyone. BITA Standards is one of most known as a standard for the trucking alliance. This data standardization must be used in order to standardize all companies avoiding different data treatment between companies.

Who validates the product?

The product prepared by a credited person, which catalogs products with a tag. All documents and papers are converted to smart contracts, which helps export customs to understand the type of product and their dimension or weight. The transparency of blockchain and their constituents like smart contracts allows checking the certificate entered by third parties and proves that the product is trusted.

All documents included in smart contracts will be shared in the ledger to improve transparency and track provenance, tracing information about the driver or truck, which contains this.DHL (2018).

A producer can also tag a cargo with a *Radio Frequency Identification (RFID)* TAG, which contains the certificate of origin, batch id, and produced date. Depending on the type of product to be exported, there are several methods to trace the quality of the products, such as an example developed by IBM (Crypto-anchor) IBM (2019).

Thus, the package can be equipped with an IOT sensor like temperature/humidity monitoring preventing damages by changing temperatures such as meats, pills, wine, and many other products DHL (2018). Another example is a simple magnetic sensor, which broadcasts an alert if the package is opened.

Smart contract can be amended by adding additional terms and conditions:

- Considerable delivery delay.
- Delays during cargo inspection.
- The client fails to collect cargo.
- On a damaged object, the client receives a refund.
- The cargo does not match.

Who adopted the technology?

RoadLaunch

Road launch is a private blockchain-based in digital freight management working with Hyperledger. This platform was created in Canada, and its propose is integrating all enterprise positions in the all-in-one platform, using modules for payment. As far as it is concerned, this platform is the most similar to the subject of this dissertation, due to the integration of the carrier and shipper. RoadLaunch WebSite

Shipchain

Shipchain is a public or private blockchain-based sloped-oriented to product traceability, that works with Ethereum. The central vision of Shipchain is to make a solution to help customers to track and trace their order. Using Ethereum is an advantage for the payment facility. ShipChain WebSite

Skycell

Skycell is an enterprise based in Switzerland with the purpose of selling containers equipped with IOT sensors. These containers are used to monitor drugs and control the disruption of the temperature range, maintain the quality of the product. All records are based on Blockchain technology in real-time. SkyCell WebSite

Maersk

IBM and Maersk have coupled up to build a monitoring solution for the cargo of ships, using Hyperledger. Present worldwide, the world's largest company leads the logistics industry. Maersk WebSite

TE-FOOD

TE-FOOD aims at providing greater transparency in the food industry and improve consumer confidence and brand exposure. TE-FOOD's blockchain (the FoodChain) is a public permissioned blockchain used as a food traceability solution from farm to consumer. In summary, TE-FOOD has several components that allow the supply chain to identify the items that have been tracked, as well as obtain the data and save it in the blockchain, next this data is processed to present it to consumers.TEFOOD WebSite

CargoX

Cargo-X is a technology that aims to allow the consumer to follow the path of their cargo from the cargo to the end of the delivery process and also visualizes the entire transport process. This technology is based on blockchain that provides a fast, secure, and reliable way to process bills of lading. This technology is also based on extensive learning of data and machines, which also allows you to analyze large amounts of data. Cargo-X has developed a decentralized platform based on the Ethereum network.CargoX WebSite

Although these technologies already offer blockchain solutions in the market, in this dissertation we decided to apply blockchain, due to the added value of making a versatile system, which uses mobile devices, to track orders. However, its application differs from other technologies by the integration in real-time systems between the various mobile devices.

Is the data exposed?

Written proofreading, promote us an idea of the various types of blockchain and which is the best method to implement on organizations, being always the doubt of method to choose. Consequently, the best way to implement for this dissertation is a private blockchain.

A user with a pseudonym like identification can interact with the ledger without revealing their true identity in public blockchain. However, there is a problem that data is exposed, not being expected for this dissertation. With private blockchain there is a total secrecy about the identity of the user.

BITA Consortium appeared in order to create a common framework to incentive the development of blockchain applications for logistics by using private transactions for the trucking industry, such as shippers carriers and regulators. Salama (2018)

Nevertheless, this implies a real dilemma about the best option to choose, for the reason that many people believe that blockchain needs to be public to satisfy their true potential.

The best practice for the trucking industry will be discussed even though the blockchain is defined as "trustless". Each company has the ability to record relevant information and choose each participant can join the network. A layer of security is another way to examine trusted entities extending that participants would be controlled by other users in network.Nichols (2019) There are other ways to conceal information, as described in section 2.11.6.

What are the advantages of migrating?

An organization that using Blockchain looks better on the market, due to the usage of new technologies and better response efficiency to requests, the case got more turned around trucking services which mean that described topics are based on this fact.

According to Lindholm from Volvo, there are some crucial aspects by adopting this technology. One of them is Tracking Asset, being already referenced in previous sections, but there is an aspect that may not have been related, which are maintenance records of the vehicle. Vehicle information cannot be unchangeable. Therefore there is a good advantage for the second-hand vehicle market.Lindholm (2019)

Another relevant point is regulatory demands and audits, these transactions are being clear and simple and simplifies auditing throughout the customs process. Sid (2018).

3.2 HYPERLEDGER GRID

Grid is a project focused only on the industry of supply chain, approved in January 2019. It is a framework that provides developers create supply chain–centric solutions based on a Distributed ledger, including Smart Contracts that are integrated into the core of the framework to combine components. Grid Track and Trace is one of the most prominent examples which aims to develop a module that knows the history of possession and characteristics from temperature to location.

One of the advantages of this solution is the implemented know libraries such as BITA consortium and GS1/GTIN. This framework as been currently in development and incubation, since April 2019. Hyperledger (2019b)

3.3 SMART CONTRACTS AND LOGISTICS

The function of smart contracts, which focus on logistics, has the function of automating processes in such a way that they do not need intervenient requirements.

According to DHL (2018) digital documents and real-time shipment, data must be supported and incorporated into the blockchain in order to support and automate the consistency of smart-contracts. Figure 10 evidences the process of automation, while both parties accept the conditions in the contract.

As we know, blockchain technologies can be used in several sectors as well as healthcare area, automotive industry, logistics and in textile sector to prevent counterfeit goods. A typical scenario of shipping goods, shippers, carriers, customs tax, consignees are involved and for every shipment, they need all documents to execute a legal and authentic process of shipping. Normally these documents are called BOLs (bills of landing) and PODs (proof of delivery) and more documents are needed depending on case type. Blockchain offers a good advantage to use this data because the network member can validate the block or payload of the transaction.Carter (2018) The efficiency and transparency on the blockchain are used to manage all documents and transactions that are used for the logistics and supply chain process.

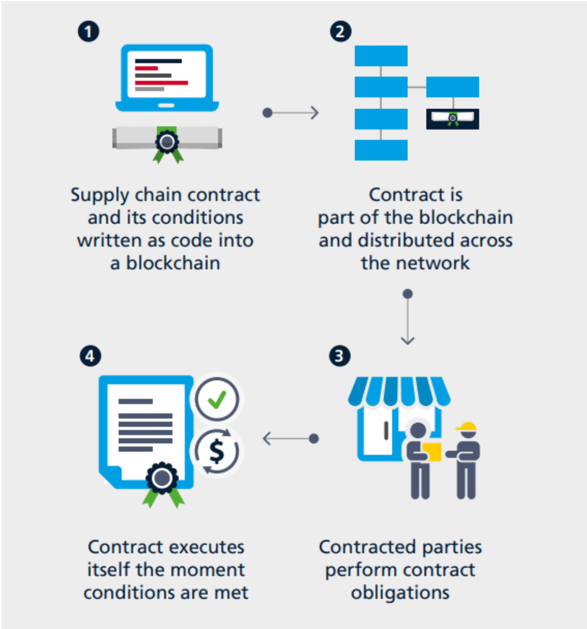

Figure 10: Smart Contracts in logistic industry , Source: DHL (2018)

3.4 BITA CONSORTIUM

Blockchain in Transport Alliance (BITA) is an organization composed by a hundred members from the sectors of logistics and transport focused on blockchain alliance, established in 2017, with the primary objective of creating a standard, especially for the logistics and transportation area, as well as encouraging the use of new solutions by companies. BITA uses the same principle of Blockchain-based on *Distributed Ledger Technology (DLT)*, enabling the participants to adopt new frameworks and standards in order to develop an innovative blockchain.BitaStudio WebSite

3.4.1 *Standards*

Recently, standards have been launched with the objective of ensuring interoperability between systems.

BiTAS Tracking Data Framework Profile

This standard has as purpose tracking a parcel, and answering the question "Where is my Shipment?", in order to answer the question, this framework profile was developed in order to determine the location quickly by showing the appropriate information. Such as detailed information on the progress of their route and operational processes until the end of the route, primarily with the purpose of focus on tracking and tracing across a transportation network. Main data structures are related to other structures in order to reference them in a form that is understandable and perceptible to the user. Kothari (2019)

Location Component Specification

This component is dedicated to investigating several localization components in order to have a uniform standard for localization tracking.

According to Soni (2019), this data structure should always be used when geographic locations are stored in Blockchain. However, this solution must always be embedded in another data structure. The documentation available provides models of components, highlighting : Track a shipping container location, Record the destination address of delivery, Record a customer's billing address for a shipment and Send a driver to a location.

3.5 GS1

GS1 is a non-profit organization created in 1974 to define standards in order to facilitate communication among business partners by highlighting suppliers, distributors, retailers, and end customers. Developing applications compliant with the standard can improve operations in logistics due to better management and cost reduction with the necessary documentation, a numerical identification is used to ensure the capture of data each time an item is movedGS1 (2019b).

GS1 is composed by three main numbering elements:

- GTIN – Global Trade Item Number;
- SSCC – Serial Shipping Container Code;
- GLN – Global Location Number;

3.5.1 *GS1 standards and blockchain*

The purpose of GS1 in technological terms refers to the adoption of the blockchain to ensure data integrity and trust between both parts in order to get a version of the events in the logistics chain. The use of standards aims to structure data and reduce their replication, in such a way that interoperability is ensured.

Figure 11 illustrates the interoperability between layers, in order to identify, capture and share data,

To understand the purpose of GS1 concerning blockchain, some aspects will be presented to understand what is stored GS1 (2019a).

1. A single event data signed cryptographically, with a focus on scalability.

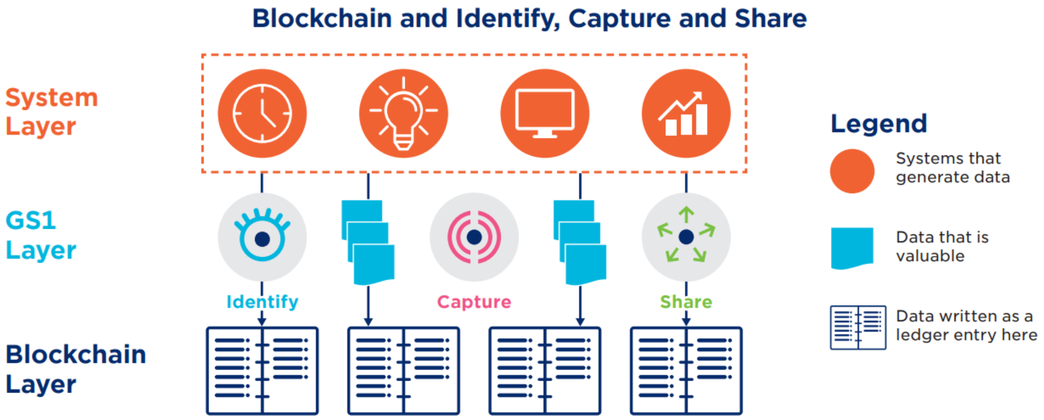

Figure 11: Gs1 Architecture to Identify, Store and Share , Source: GS1 (2019a)

2. Off-chain data, through a separate tracking application, comparing hashes so that the information has not been altered.

3. A hash with data pointing to the off-chain data. With this suggestion, it allows the ledger to stay in discovery mode, for participants who want to share data and communicate.

3.6 Summary of Blockchain in Logistics

After the study of the main concepts and approaches in the area, as it was presented along this chapter, some conclusions about active monitoring can be drawn. To understand the main requirements, it will be necessary to adapt the methodology to the various companies in the sector and understand aspects of all intervenients in the logistic sector.

Some applications have already been created in this sector, according to market specifications and requirements. The adoption of standards allows companies to communicate with each other aiming at assuring system integration and interoperability.

Companies are still using old systems such as *Electronic Data Interchange (EDI)*, and the associated documentation, taking a long time to dispatch and import orders. In the point of view of the market, the administrative areas must begin to show the benefits of blockchain, in order to experiment and report respecting the requirements that are necessary for the subscription of this technology.

The applications already available offer competitiveness compared to the conventional model. RoadLaunch is mentioned in this dissertation, due to its purpose for road transport, and dispatch of orders between one point and another, integrating several companies from the same sector into a single application, not meeting the door-to-door tracking.

However, according to the aim of this Master's work, a system is desired that can be used to monitorize drivers, routes, and all orders from its dispatch to final delivery.

4

PROBLEM MODELLING AND SYSTEM SPECIFICATION

After the state of the art review, there was a need to improve the efficiency and consistency of the system, compared to what is currently used. Blockchain nowadays is a topic that is discussed, but still in constant development. So, companies have doubts and feel insecurities concerning the migration towards this new technology.

The logistics sector involves numerous operations, modes of transportation and routes, for an order to go from a starting point to a final destination. The most popular type of transport covers planes, boats, and trains, and in order to communicate among these types of transport, data must be represented in a standard format so that they can interchange information and interoperate. Documentation that must be shared along this process chain depending on the country of importation, for example, in customs authorities, it is always necessary to know what the order contains. Using Blockchain, at least, the destination identity is always informed as well as the type of order.

In this Master's work, it is aimed to use new technology in order to improve the guarantee of product conformity to the end customer. Therefore, in the logistics sector partners have to be connected to each other in such a way they can communicate and standardize data among them. However, due to a large number of participants involved, it was necessary to adapt this project to the terrestrial environment, more focused on road transport. The road transportation is a key point in logistics, for the following aspects:

- The order is delivered to the customer by this way.
- Delivery time is essential and crucial.
- Guaranty that the product delivered to the customer is authentic and satisfies the requirements.

After this introductory analysis, next section identifies the essential intervenients and enumerate the different requirements that must be observed in the development of a solution.

4.1 REQUIREMENTS

The system operators aim at supporting user and order management functions, while the user, chauffer, and customer operate and have direct interactions with the system. All intervenients are essential in order to support what was defined.

System Requirements

Requirements	Type	Importance	Complexity
The system should allow the admin to manage enterprises	Functional	High	Medium
The system should allow the admin to manage orders	Functional	High	Medium
The system should allow the user to manage orders.	Functional	High	Medium
The system should allow the user to verify truck position.	Functional	High	High
The system should be able to determine weight of freight	Non functional	High	High
The system should be able to alert user about maximum weight and capacity of truck	Functional	High	High
The system should show notifications about truck maintenance	Functional	Low	Medium
The system should be able to determine delivery time	Non functional	High	High
The system should inform the user about duration of each order.	Functional	Medium	Medium
The system should inform the customer about shipment status	Functional	Medium	Medium
The system should be able to inform the chauffer about ordering of orders	Non functional	Medium	High

User Requirements

The user can manage chauffeurs	Functional	High	Medium
The user can manage trucks	Functional	High	Medium
The user can create a shipment	Functional	High	High
The user can view and manage all shipments in real time.	Functional	High	High
The user has ACL permissions	Non-functional	High	High
The user can alert a chauffeur about updating on orders	Functional	High	High
The user can verify all integrity of data about the orders.	Functional	High	High
The user can update the status of order	Functional	High	High

Chauffer and Truck Requirements

The system must update truck location	Functional	High	High
The chauffer must be fit for emergency mode.	Functional	High	Medium
The chauffer must be able to inform customer if something wrong.	Functional	High	Medium
The chauffer can update status of order.	Functional	High	Medium

Customer Requirements

The customer must be able to follow the order in real time	Functional	High	High
The customer must be able to sign electronically	Functional	High	Medium
The customer must be able to receive a Tracking ID.	Functional	High	Medium

4.2 MODELS

After the elicitation of requirements and before the system implementation, it is mandatory to model the desired software. This section will present the models built.

4.2.1 *Uses Cases*

Use cases have the function of defining interactions that occur between the system and the user. There are two types of system representation, business actor, and business uses cases:

Business actors Operations Center can be admin of System, Customer, Chauffer, all performing system operations.

Business uses cases by highlighting "Manage shipment" which includes "Create shipment," extending "Confirm shipment" and "Update location".

Figure 12 illustrates the use case applicability.

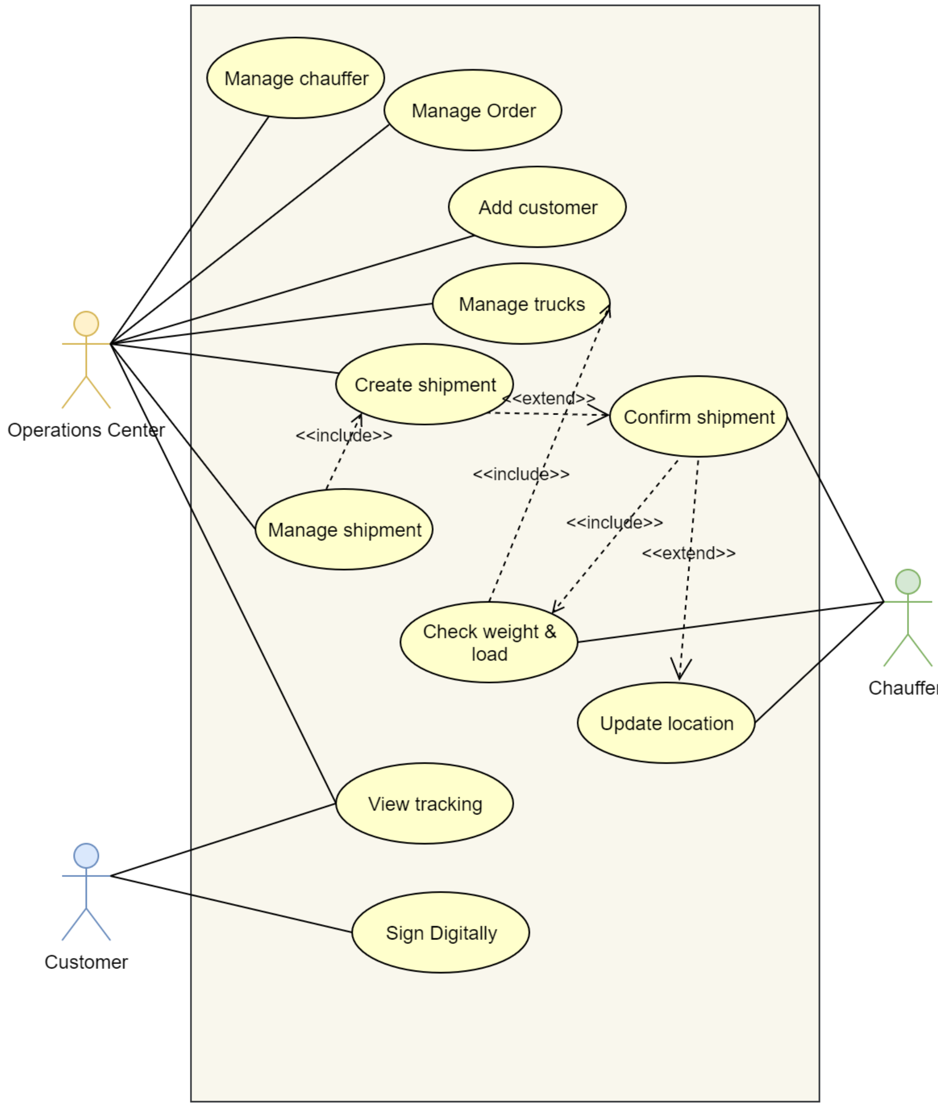

Figure 12: Uses Cases

4.2.2 *Sequence Diagram*

The sequence diagram is focused on users who interact with the system. This scenario demonstrates an incoming order, dispatch to the shipper, and delivery to the end customer. It contains system interactions that highlight when the manufacturer receives information about the order and on final delivery when the customer receives and confirms the order.

It will be decided to centralize the diagram on order dispatch due to numerous interactions with the system when order is being sent, and also the delivery to the final client by confirmation with proof of delivery.

Figure 13 demonstrates a real application scenario of this system.

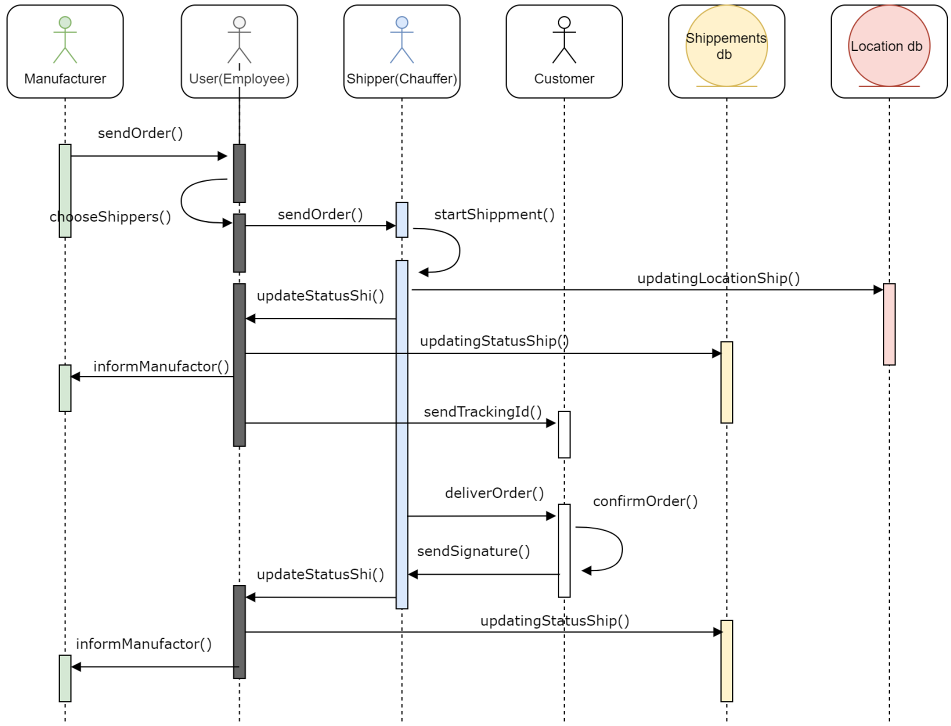

Figure 13: Sequence diagram (Order Dispatch)

4.2.3 *Entity Relationship*

The relational entity aims at defining the relationship within entities. This model, referred in Figure 14, evidences data infrastructure that the system can support.

As we can analyze the relationships that exist between the entity "Shipment" are the actions that include more data support, due to being the main component used in the system.

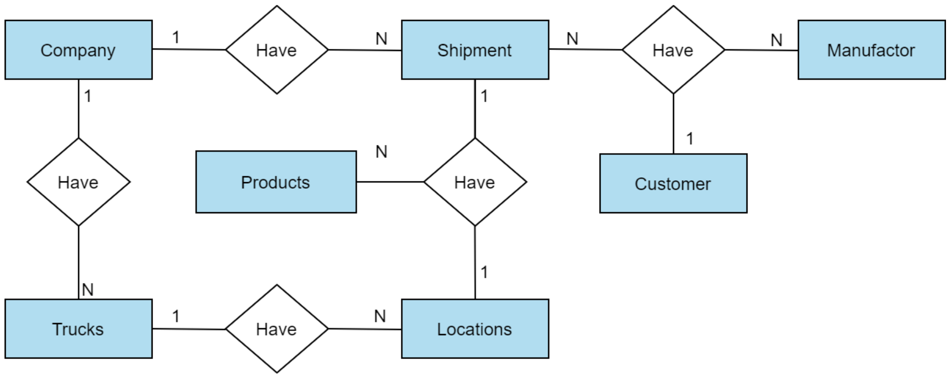

Figure 14: Entity Relationship

4.2.4 *Database*

In this organized data collection, the system should support the requirements referred on section 4.1, in order to support what was mentioned, the diagram mentioned in Figure 15 provides a visual overview about relationship between tables.

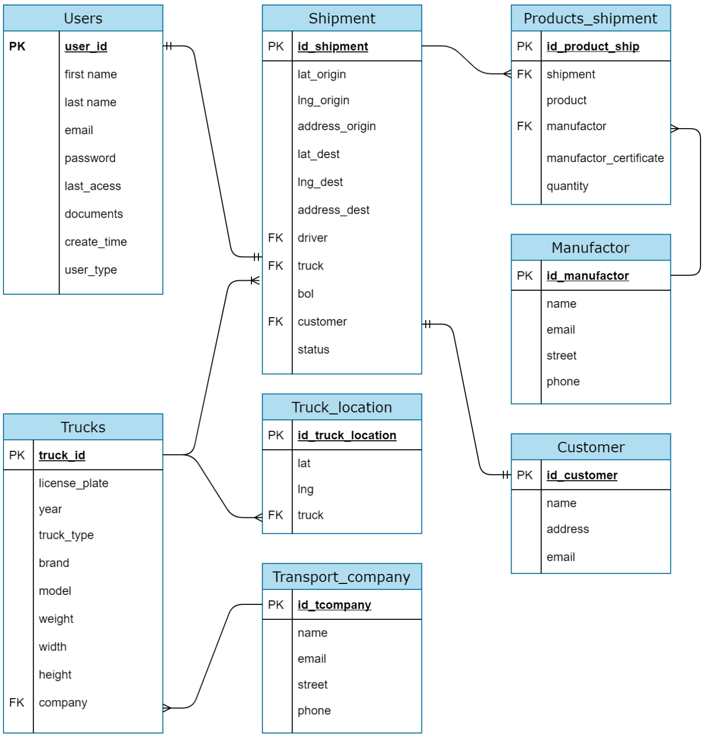

Figure 15: System database

4.3 SUMMARY

In a general overview, the system was modelled in order to support the data specified in the requirements, and yet, it is considered the amount of information that the system can support in terms of scalability, performance, and response time.

However, the system must be focused on continuous monitoring and response time must be short, and precise. The data stored in web servers must be specified and designed in order to prevent system failures, by avoiding system overload, and secure data with regular backups,

5

TECHNICAL IMPLEMENTATION

5.1 HYPERLEDGER PLATFORMS, AN OVERVIEW

To understand the requirements involved in this Master's work, some frameworks can be excluded due to their type of application. Three recommended frameworks will be presented which can be implemented into this dissertation, these frameworks support SCM, although there are advantages and disadvantages due to its features. It will be decided to focus on crucial issues such as privacy and trustworthiness of transactions since one framework is permissionless, and the other supports both. Table 5 shows the main characteristics and what purpose is suitable for the development in the real context.

Framework	Consensus	Type	Smart-Contract	Language	Application	Status
Burrow	BFT	Permissioned	Application Engine	Native, Solidity	+ 200 transactions per second	Incubation
Fabric	Kafka, Solo, Rafta	Permissioned	Chaincode	Go, Javascript	Supply Chain, Finance	Active
Grid	None	Permissioned	AEC(Sabre)	None	Supply Chain	Incubation
Indy	RFBT	Permissioned	None	Native, Java, Python	Identity	Active
Iroha	BFT, (Sumargi), YAC	Permissioned	Chaincode	C++	Finance, Insurance, Cross-Chain	Active
Sawtoth	POET	Permissioned, Permissionless	None	None	Supply Chain, Track Trace	Active

Table 5: Types of Hyperledger frameworks

After analyzing these frameworks, Hyperledger Grid is suitable for development in real context because it offers many standards natively implemented such as GS1, Open Data Initiative, and BITA Standards. The sample applications offer to Track and Trace app which is an added value for product tracing. These frameworks are under development and are expected to be operational in 2020. Consequently, two powerful platforms are Hyperledger Fabric and Hyperledger Sawtooth. Some fundamental characteristics presented that are decisive in the choice of the following platforms.

Hyperledger Fabric uses MSP a robust authentication mechanism. This MSP identities contain a CA, which uses the X.509 Certificate to define the different levels of the user, which exists in an organization. The trusted CA is needed for every member to enter in the network and make transactions. The MSP and CA are one of the significant differences compared to sawtooth because the authentication is managed by combining a large number of validations. The Sawtooth can be public, which makes it more vulnerable and can proliferate.

The channels and private data collection helps Fabric to protect data and transactions by other entities. Each organization can operate in one or two channels, which supports a different transaction per channel, ensuring privacy. Another way, with the private data collection, all organizations have the information, but the majority of this information is hidden. On Sawtooth, the *Private Data Objects (PDOS)*operates with Intel Sgx, which provides execution verifiability, and contracts can run off-ledger(on SGX-capable machines)Mic Bowman and Andrea Miele and Michael Steiner and Bruno Vavala (2018).

Regarding the consensus, both frameworks are different because Fabric uses Kafka or Raft, actually Raft, and Sawtooth uses*Proof of Elapsed Time (POET)*, which is an advantage for Fabric due to being more robust than the Rafta. Regarding the storage state, the Fabric uses CouchDB (JSON Objects) and are required to know the current state of the network to run the consensus protocols.

In general, the Fabric has better advantages over Sawtooth due to the security and privacy that exists on transactions, due to the fact that is permissionless being a crucial factor in network access control, in business scenario it allows to keep a subset of data privacy to a subset of participants, and we think this is a fundamental factor to implement on this dissertation. As far as development is concerned, the level is faster because there is a lot of documentation and examples on the official page that are easy to understand.

5.2 CHOICE OF A SPECIFIC PLATFORM

After exploring the various platforms that currently exist, it turns out that blockchain makes the difference concerning supporting services such as in logistics, supply chain, and manufacturing. The blockchain has been growing in recent years, and the concepts have been increasingly adopted and solidified by companies.

Most of the current platforms are becoming obsolete due to the evolution of the market and new technologies that will emerge, such as Hyperledger frameworks.

Table 6 evidences technologies from various business sectors will be presented. During the research, the adoption of Hyperledger frameworks was more suitable than Ethereum. The differences between the platforms are relevant; while Etherum is public, Hyperledger is private, which rules out the possibility of being used in this dissertation.

Platform	Ethereum	Hyperledger Fabric	Ripple	OpenChain	Quorum
Issue date	Jul 2014	Dec 2015	Aug 2012	Apr 2016	Nov 2016
Industry focus	Cross-Industry	Cross-Industry	Financial Services	Digital Asset Management	Cross-Industry
Permissioned	No	Yes	Yes	Both	Yes
Smart-Contract	Yes	Yes	Yes	Yes	Yes
Smart-Contract Language	Solidity	Go, NodeJs	C++, JS	C#, JS	Solidy
Consensus	PoW	Solo, Kafka, Raft	XRP Ledger Consensus Protocol	PoW	Raft, BFT, POA
Currency	ETH	None	XRP	None	ETH

Table 6: Comparison of existing blockchain platforms

Concerning the consensus mechanism, the differences are notable since Ethereum uses POW, which means much energy is consumed, and the probability of an attack is increased, due to being open to the public.

A new consensus mechanism has been adapted on Hyperledged Fabric since May 2019, and the blockchain network has been fully decentralized after that. In Fabric, the orderer is a consensus mechanism that is responsible for running communication service and ensures the consistency of the Blockchain (delivery guarantee for all nodes). As referenced in section 2.11.1, Ethereum is more used for financial applications, which turns on a powerful way to manage the money of the users.

To conclude, the new consensus mechanism recently added on Fabric demystifies the fact that it is not fully distributed as in the Kafka mechanism, as long the Fabric is recommended by its privacy guarantee and examples of applicability in companies. The transactions recorded in the blockchain can be easier to interpret by developers, with this, the existing libraries make this platform more user friendly in such a way that can be deployed in various environments. The pluggable consensus on this platform allows to change the mechanism at any time, presumably in future a BFT based consensus mechanism is expected to be applied, which makes it more robust.

5.3 Implementation overview

After a comparison made in Section 5.2, the framework that corresponds to this dissertation is Hyperledger Fabric because of its strong authentication and viability mechanism. However, along this chapter, this technology will be further explored in order to understand its mechanism and implementation easily.

Through these observations and focal points, the purpose of this implementation, is improving daily technology compared to what already exists on the market, make provision for efficiency and innovation, security on data, implement new business strategies and improve customer confidence. Maintaining what is considered the good aspects of today's technology, the new proposal that will be presented, aims to implement a solution that serves a transport company's needs by using Blockchain.

Taking into account the aspects mentioned above, Figure 16 sketches the architecture used in the blockchain-based system.

5.4 Blockchain architecture description

The architecture presented in Figure 16 demonstrates a basic system using Hyperledger with Raft Orderer. Orderer, as referred in section 2.11.6 is defined as Consensus in Hyperledger. Through these, the system architecture is based on three fundamental elements, such as Manufacturer, Regulator, and Shipper. These elements are designated as organizations in which each one is regulated by the MSP, being required for managing user authentication, and the organization.

Each organization contains a database designed as CouchDB to act as a world state database, according to Hyperledger (2019a) it is executed on the separated process by the side of peers, supporting *JavaScript Object Notation (JSON)* document storage which allows us to perform querying operations.

This architecture presented is composed by two parts: the first designated as A shows, a simple scenario, one customer purchases an item from the manufacturer, and the entity will be responsible for managing purchases and forwarding it to a shipper. After the shipper initiates the route, the manufacturer is informed. Since then, a first proposal transaction is initiated on Hyperledger, and a tracking number is sent to the customer.

After the first transaction, the system is able to send location information via mobile phone and executes them by inserting them into the Ledger, keeping a ledger up-to-date and immutable.

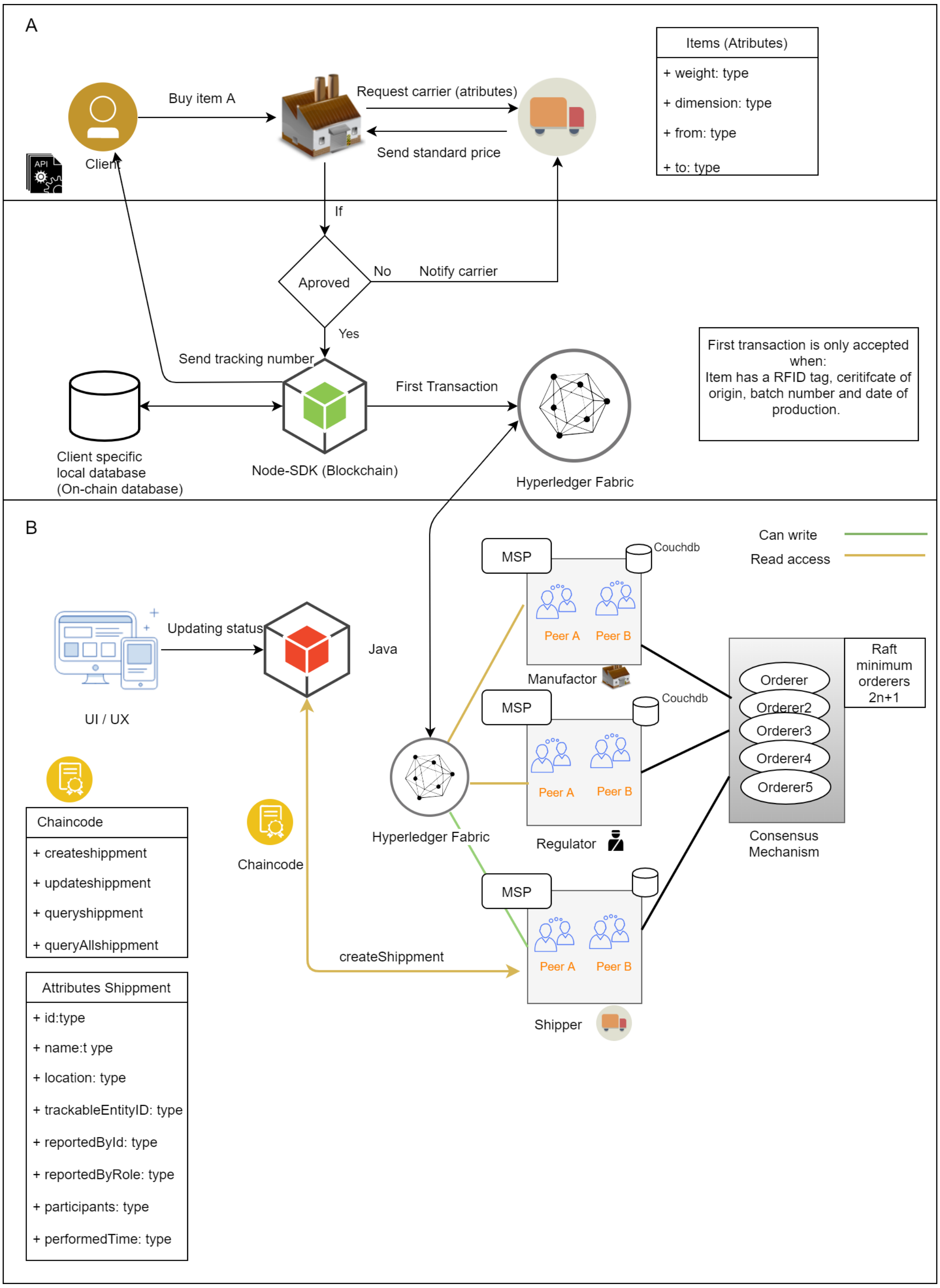

Figure 16: Blockchain Logistics Architecture Scenario

5.5 SYSTEM ARCHITECTURE DESCRIPTION

This description presents us with an abbreviation of the system in question involving the three main intervenients. These three participants have as their function to make network requests in order to perform all transactions in the network. This diagram mentioned in Figure 17 evidences the sequence of requests made and the possible response from the server.

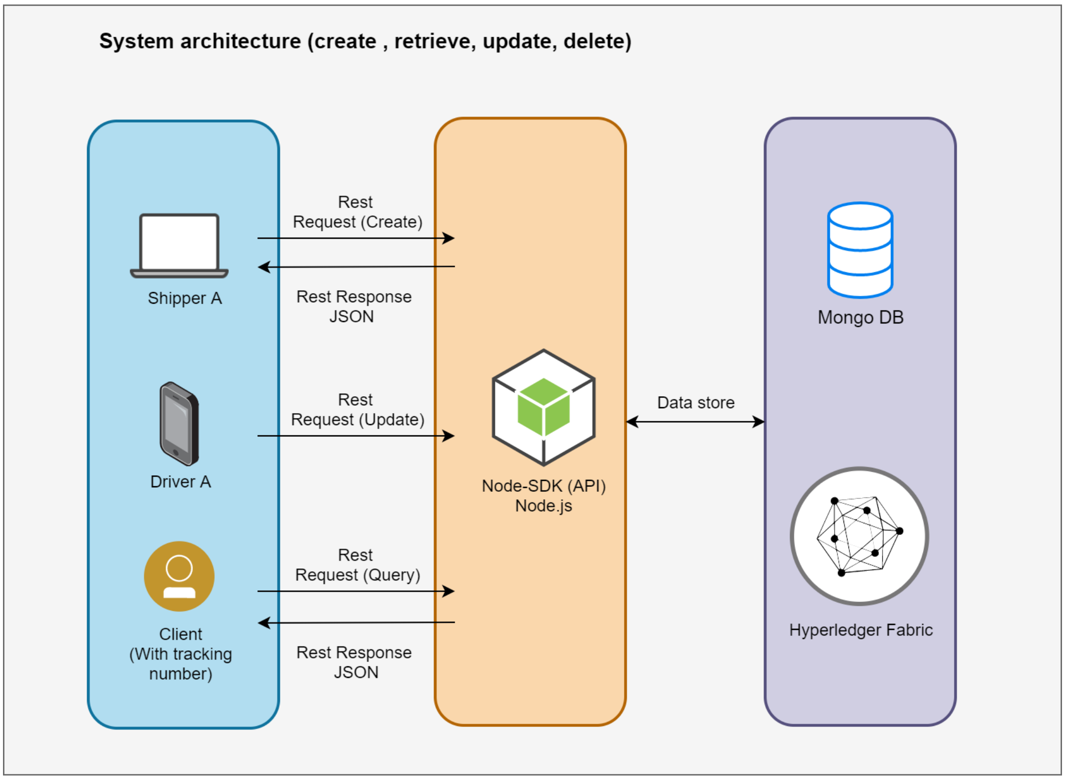

Figure 17: System Architecture (CRUD operations)

A simple system description can be made according to Figure 17: A Company receives an order from a manufacturer, and then try to dispatch it according to the number of requests. Before sending, pieces of information must be inserted on the system such as Type of order (which must be cataloged with a RFID Tag), origin, destination, customer, bol if applied, Truck and Driver. After the information is submitted, the Driver receives a warning on the smartphone, with information about the cargo and the destination. When the Driver starts the distribution, the customer receives the tracking code, and the manufacturer is notified. From this moment on, all transactions are made in the Blockchain, with information regarding the order and location. After the order arrives at the customer's home, a signature is required for validation, when confirmed, the central office receives a warning, and the manufacturer is notified again.

5.6 EXPERIMENT SETUP (SOFTWARE ARCHITECTURE)

The software architecture presented in Figure 18 is split into three layers: First, the Presentation Layer, described as *User Interface (UI)*, the access can be via web or mobile. On the web interface, the user makes actions at the level of user management, shipping, vehicle management, and query the shipping status already in the mobile interface are actions at the driver's level such as, accepting the order, and sending the location to Blockchain.

Second, *Business Logic Layer (BLL)* , is the middle layer of communication between the UI and the *Data Access Layer (DAL)*. In this layer, all requests to the database and Blockchain are executed through the Express JS. This framework assists in organizing the application into a *Model-View-Controller (MVC)* architecture. The authority certificate has the function of verifying which user can access the Blockchain in order to make requests. Third and last, is where the data is stored.

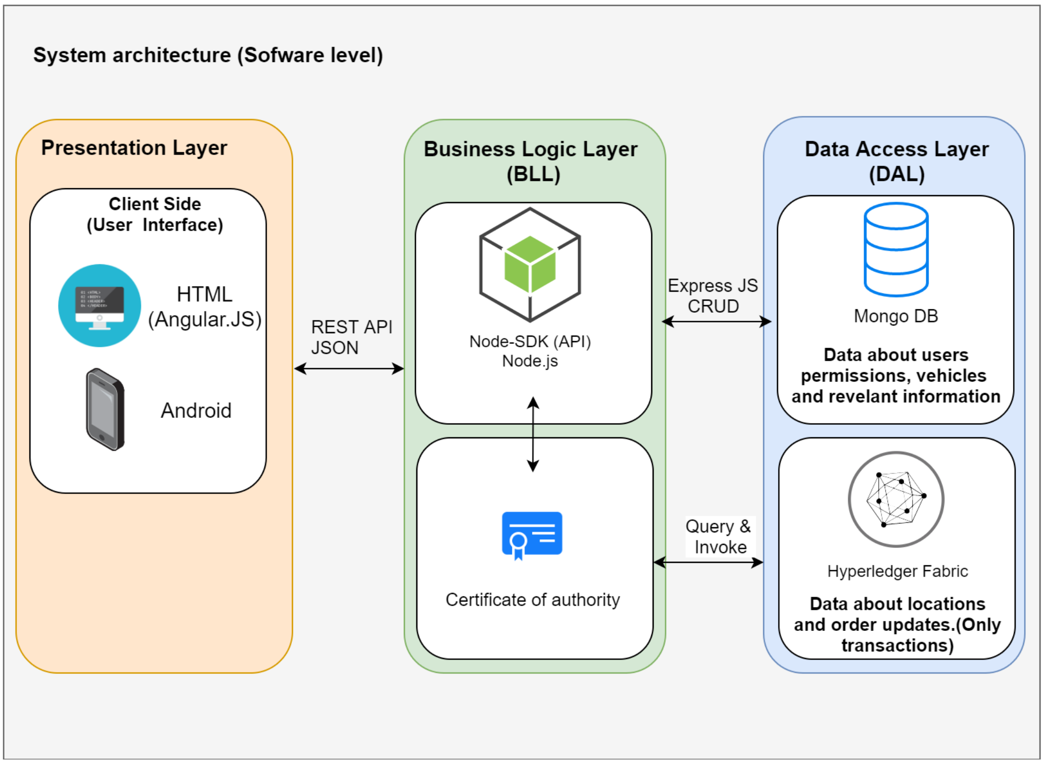

Figure 18: System Architecture (Software level)

To better understand this system architecture, the following sections are specifically explained, how data is presented and sent to the different layers of software.

5.6.1 *Front-end setup*

In this subsection, the components will be presented where interactions between users and the website occurs. As mentioned in the previous section, all information visible to the user is demonstrated on the front-end. The platform used for this purpose is *Angular 7*, where it uses a different design pattern rather than the one currently used.

Operating with modules and components, where a module is a piece of code designated for a single purpose, and with this perspective, it is applied to this division modules that are assigned to the management of the main system tasks. In the interaction with the user, the login, user session, and multi-language are highlighted. Whenever a user logs on, a *JSON Web Token (JWT)* component generates a token with a duration of 1 hour (session time), that token is required for all system interactions, highlighting database query.

Compared to the mobile version, there are a few contrasts due to its exclusionary driver-oriented. Android-based, the interactions that exist between driver and application is the acceptance of the shipment, and a route to the final destination.

5.6.2 *Back-end setup*

Back-end, in our perspective, is responsible for entire system operations. It contains the fundamental components for the interaction between the Front-end and Back-end, highlighting the requests to the database. The responsible framework is *node.js*, which is designed to migrate all the programming involved in the front-end to the server.

It is decided to use this technology for its flexibility and component support. The structure of this component is divided into the following modules: Models, Controllers, and Routes. The models are distinguished by their connection to the database *(MongoDb)* and for the creation of collections (means the same as tables in a conventional database). Controllers are responsible for all system functions such as requests to the database and the return of information in JSON, and lastly, the routes, which aim at pointing the server's path in the front-end (browser) in order to request data. In this back-end, the main features are highlighted: Token generation through JWT, the function that allows the user to make requests to the database if logged in, creation of salts, for previous password recovery, function for the generation of tracking identification and the mail server.

5.6.3 *Hyperledger First Setup*

With the review of the background and documentation support, it will be decided to focus this dissertation on this platform. As mentioned in section 2.11.6, this platform is pluggable where the user can adapt the technology to their needs. Through this, in the following section, it is will highlight the components that were used.

Containers and Components

Across this platform, a previous structuring of the network is necessary. As referred in section 5.4 and on figure 16 the network topology presents some complexity.

The software used for network emulation is named *Docker*, which is responsible for creating separate containers, an example, simulate several machines in different networks and create a channel between them. With this methodology, the following figure 19 describes the components that have been installed in order to perform operations between the various containers.

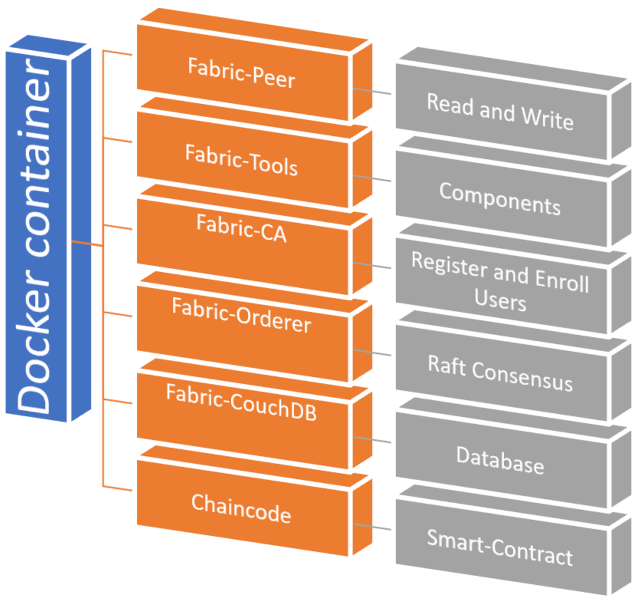

Figure 19: Docker container structure Adapted from: dos Santos et al. (2018)

After research and various experiments in order to generate the network, it was necessary to download files from the source of *Hyperledger v1.4.0* and then create the necessary files for network configuration. These include "crypto-config.yaml", which is responsible for setting up the organizations, "Docker-compose.yaml" that is able to associate each peer to the network with dependencies of the *CouchDb* and the *Orderer*.

The following steps describe how the network is generated:

1. Organization creation.

2. Generate channel and consensus mechanism.

3. Associate an anchor peer to an organization.

Chaincode

Regarding the chaincode, we can affirm that it is the main component of the *Hyperledger Fabric (HLF)* due to every operation in the network passing through it. Following the review related to the standards presented in section 3.4.1, a Standard is used in order to ensure interoperability between systems. The programming language used for chaincode development was "Go". An example presented on figure 20 represents the data structure used in the chaincode.

```
type LocationUpdate struct{
    Id string `json:"id"` //trackableProduct
    Name string `json:"name"` //Updating Location
    Location string `json:"location"` //lat, lng coord
    TrackableEntityID string `json:"trackableentityid"` //id product(!)
    ReportedBy string `json:"reportedby"` //id driver
    ReportedByRole string `json:"reportedbyrole"` //id truck
    Participants string `json:"participants"` //Carrier,Manufactor
    PerformedTime string `json:"performedtime"` //horas

}
```

Figure 20: Chaincode Data Structure

Functions were developed with the purpose of initializing the Ledger and adapt the queries to the needs involved in the requirements. Figure 21 is an excerpt of principal components on chaincode, highlighting a request to the Ledger of the locations on *args[0]*, and the insertion of a new location.

```
//insertLocation
func (s *SmartContract) insertLocation(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    if len(args) != 9 {
        return shim.Error("Incorrect number of arguments. Expecting 9")
    }

    var locationUpdate = LocationUpdate{Id: args[1], Name: args[2], Location: args[3], TrackableEntityID:
    args[4], ReportedBy: args[5], ReportedByRole: args[6], Participants: args[7], PerformedTime: args[0]}
    locationUpdateAsBytes, _ := json.Marshal(locationUpdate)
    err := APIstub.PutState(args[8], locationUpdateAsBytes)
    if err != nil {
        return shim.Error(fmt.Sprintf("Failed to insert location: %s", args[0]))
    }
    fmt.Printf("-Location updated on :\n%s\n", locationUpdate)
    return shim.Success(nil)
}


func (s *SmartContract) queryLocationShipment(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    if len(args) < 1 {
        return shim.Error("Incorrect number of arguments. Expecting 1")
    }
    Id := args[0]
     queryString := fmt.Sprintf("{\"selector\":{\"id\":{\"$eq\":\"%s\"}}}", Id)
    queryResults, err := getQueryResultForQueryString(APIstub, queryString)
    if err != nil {
        return shim.Error(err.Error())
    }
    return shim.Success(queryResults)
}


func getQueryResultForQueryString(APIstub shim.ChaincodeStubInterface, queryString string) ([]byte, error) {
```

Figure 21: Chaincode Excerpt

The ”queryLocationShipment” function has a particularity by submitting a direct request to *CouchDB*. Therefore the data returned are of the type *JSON document store* capable of performing rich queries based on keys. After the conclusion of the chaincode it is necessary to install on peers in order to confirm if there were any programming errors.

Preliminary challenges

There were some challenges during the implementation of the HLF with the emphasis on the creation of the topology , and the starting up the network. However, we have seen the need to create a virtual machine, for possible mistakes, in order to delete and clone a new image for testing.

5.6.4 *REST API*

With the data involved in the implementation process, it was necessary to deploy two *REST API*. Both of them have the function of making requests to the database through the ”node.js”. Concerning the first API, the data presented are relative to user authentication, and the submission of a shipment to the final client, among them, it will be explained the following requests:

***POST** Login - http://localhost:3000/api/authenticate*

This request is shown in order to enable the interaction between the two platforms. When executed with the required variables, returns information about user validation in an example, password error, invalid e-mail, unregistered use. All this information is returned in JSON format, then, each time the user is validated, a token is returned, which contains hidden pieces of information about the user id.

***POST** New shipment - http://localhost:3000/shipment/*

This request has the function to perform operations in the database. All requests of this type are relatively similar, return messages about success and error due to server-side validation. We have implemented an authentication mechanism to be able to make requests to the database, with this, the requester is always recorded.

The second API presents data regarding the transactions that exist in the blockchain. It will be considered that it is necessary to adopt two communication ports to avoid creating conflicts between the existing computer and the virtual machine.

***POST** Insert location on blockchain - http://localhost:9199/insertLoc/*

This request is only made by the mobile version, and its purpose is to insert the tracking locations on blockchain during the order delivery.

***GET** Order location - http://localhost:9199/getlocbyid:id/*

Finally and fundamentally, the purpose of this request is intended to return the order location, with an authentication mechanism, ensuring control access.

5.7 RESULTS

5.7.1 *Front-end interactions*

Upon deployment, the results exposed are derived from the return of the main application functions. To perform a stable and interactive system, the necessity of integration of several systems and communication between them will be examined . It is considered the need for separation of the content in order to understand how the mechanisms work. The primary system function where there is interaction between the user and the application is referenced in figure 22:

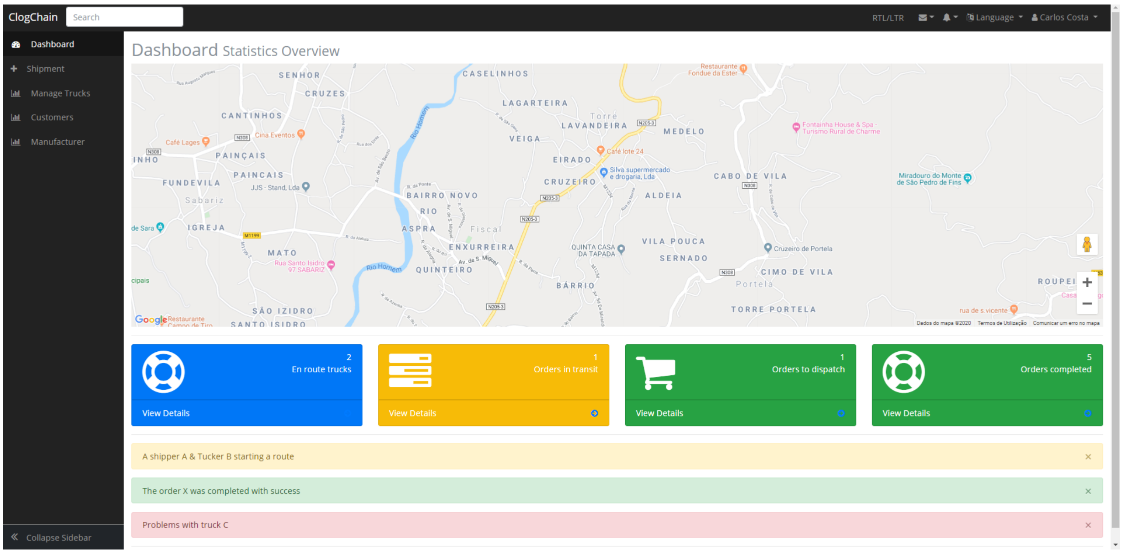

Figure 22: Main System

Concerning this, the user can get an overview of the system, with access to information regarding the location of the trucks on the map, and quantity data about the orders in progress.

In terms of shipping an order, a demonstration of how to send an order via a shipper is presented in the following Figure 23 .

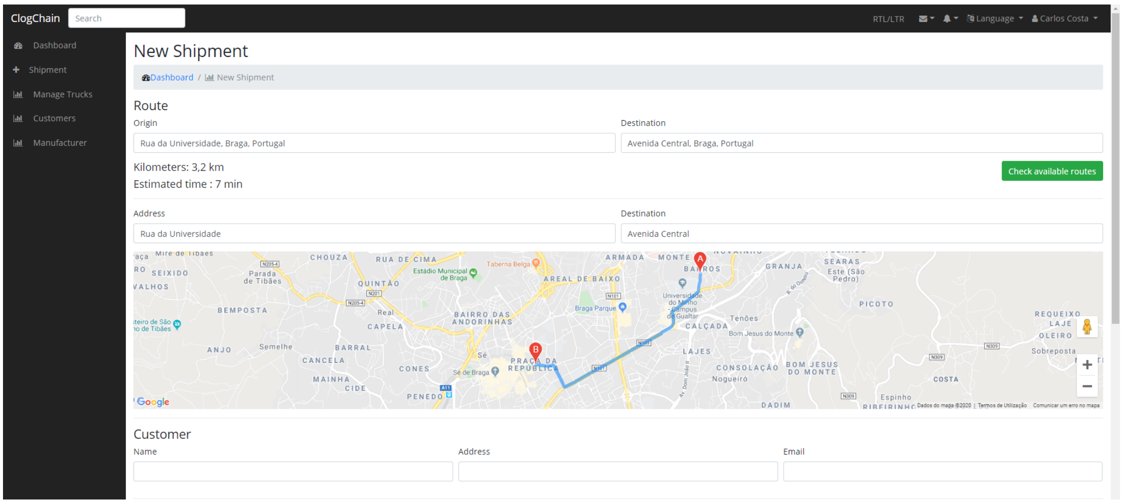

Figure 23: Submit a new order

Regarding Figure 23 we can verify the various information inputs required, since the user has the ability to calculate the desired route for the driver, and estimates about kilometers and approximate time.

After a user inserts a new shipment on web page based a new order is submitted to the back-end. Figure 24 presented the front-end interactions on mobile application, the driver has the ability to see all orders assigned to him, and are able to start a route.

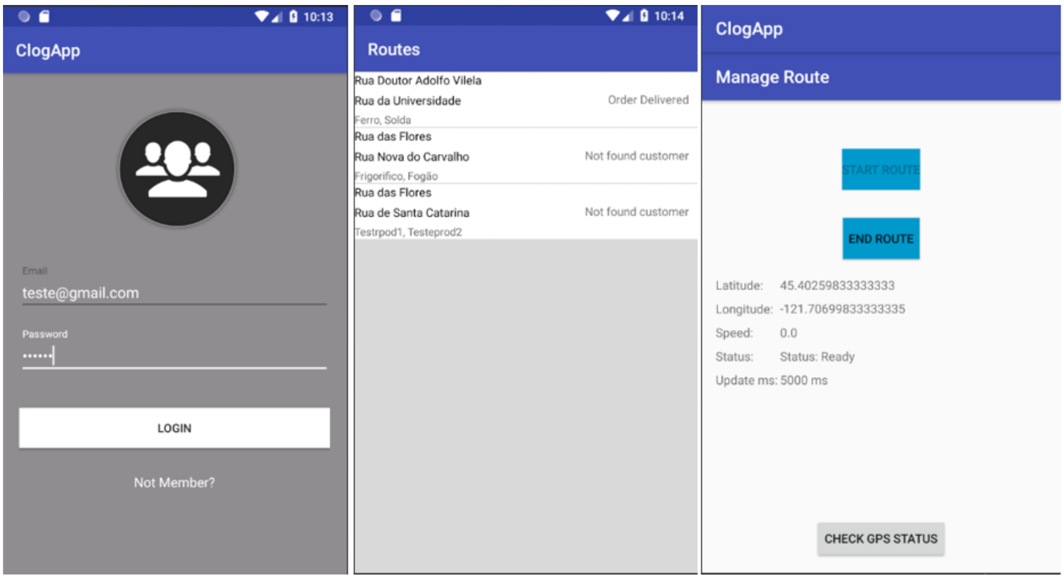

Figure 24: Mobile application

In the following Figure 25 a webpage is presented focused on the customer, here all the information related to the order status is presented in real-time, and are retrieved from the blockchain, with an update interval of 20 seconds.

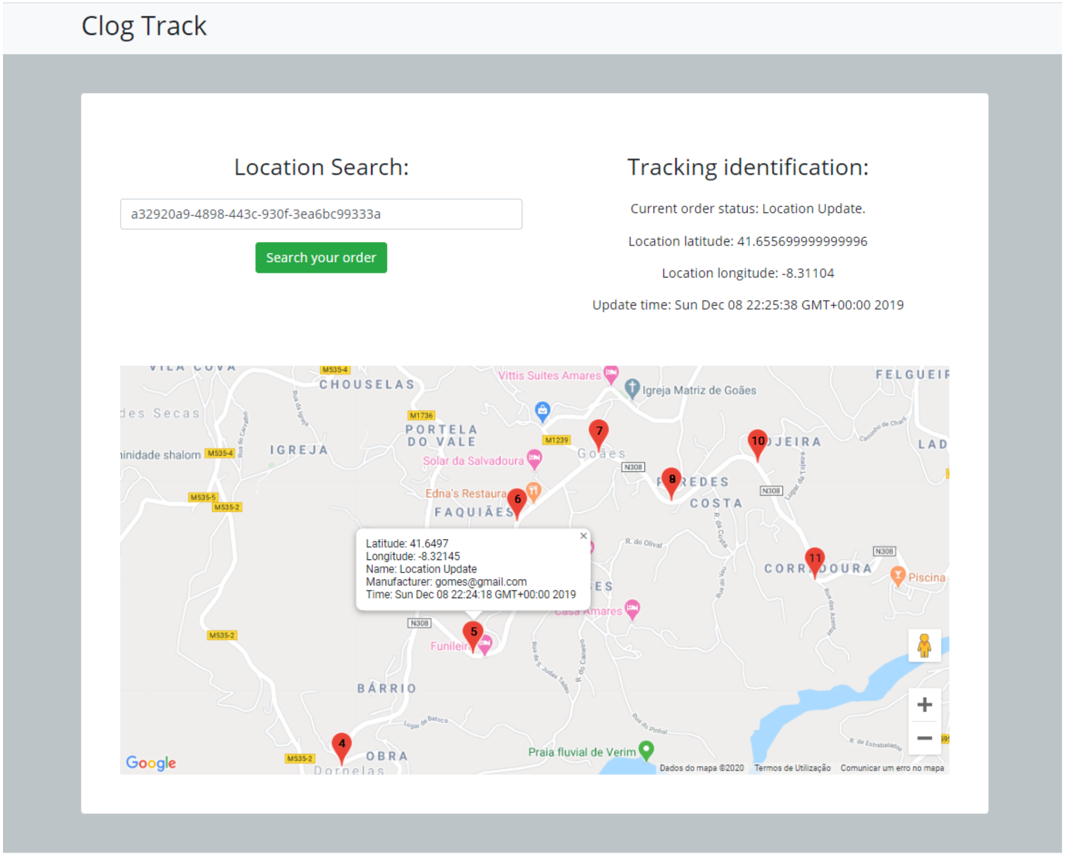

Figure 25: Tracking webpage

5.7.2 *Back-end interactions*

The results from the back-end concern on database access and interactions with the ledger. It will be decided to focus in authentication security and regarding following Figure 26 demonstrates an authentication and return of the token.


```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZTMxZTVmYmMyMTcyNmQ5ZDRiMWQzMmMiLCJpYXQiOjE1ODAzOTgwNDcsImV4cCI6MTU4MDY5ODA0N30.skOH_
  Tk9pQWChH372fb4iY8qUSLkG4Lo__J1zJ3MT4M"
}
```


Figure 26: Token authentication

In order to check the authenticity of the token, the encode and the decode on the official page of the JWT are checked, and the following figure 27 is a proof of its authenticity.

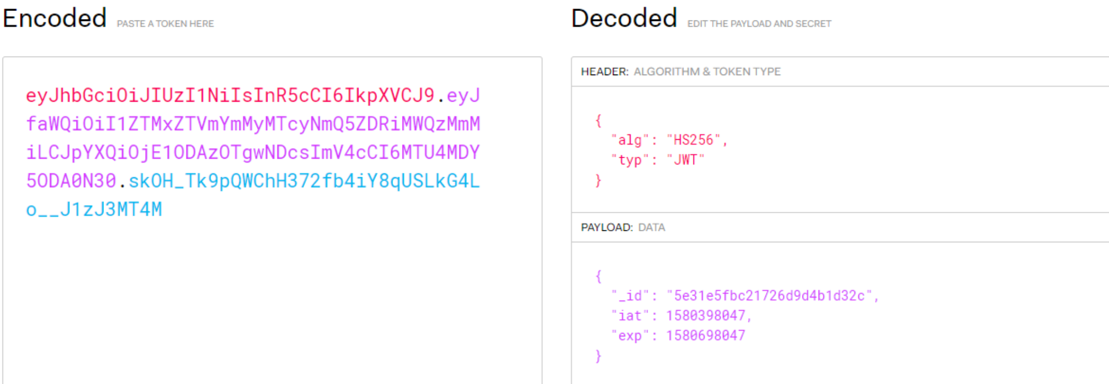

Figure 27: Token authenticity

To ensure the integrity of the token and user login information, the *_id"* presented on Figure 27 must correspond to the *_id"* which is stored in the database , the referring image 28 proves the reliability of the data.

Figure 28: User id proof

The second API shows the entire results on Ledger. The interactions between blockchain and user are demonstrated in Figure 29 that is being presented directly on the virtual machine.

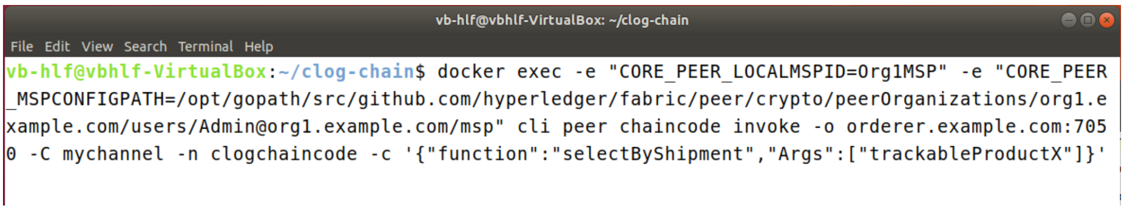

Figure 29: Query on Hyperledger

Compared to the first API the model of data presented on figure 29 is relatively different according to the data response, parsing information to JSON requires thorough analysis of the content, in such a way that data presented is correct.

Figure 30: Query result Hyperledger

The results presented in figure 31 prove the integrity of the data and its structuring are being associated to the resulting queries directly from the virtual machine.

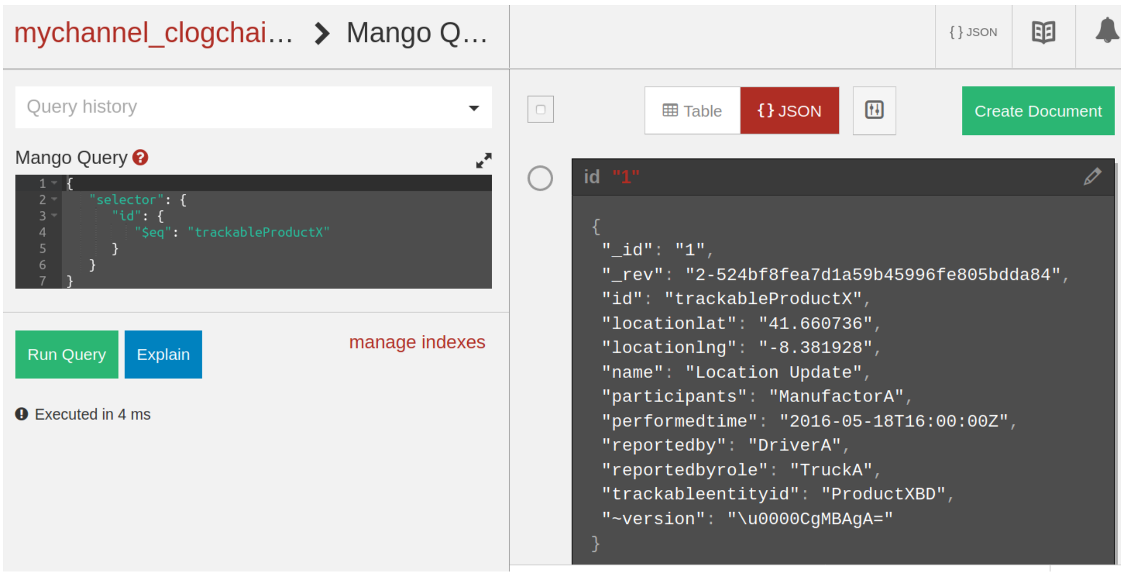

Figure 31: Query result CouchDB

5.8 DISCUSSION AND SUMMARY

Database Response

About the technologies on the market today, there is a great diversity in terms of user requirements. The response time is essential for the intended results of this dissertation, so some aspects of response time versus conventional technologies will be related. Concerning the execution response time on *MongoDb*, we can consider it to be one of the most powerful tools on the market for *NoSQL* databases.

Regarding the technology to be used, it is considered important the implement in *NoSQL*, because it is a relatively new tool in the context of databases, being much requested and adopted by today's technologies. Since the software uses a base structure document, it considerably facilitates access to data and contributes to the scalability of the dissertation development. With this, the creation of additional tables is no longer necessary, for example, the association of a product with a shipment as referred to in Section 4.2.3. Requests to the database based on "key-objects" are a fundamental factor of response time in the context of the program, due to the return of objects that contain only those types of keys.

Market analysis

The present technology is an innovative solution that promises cost savings for organizations. From this, the availability of using reachable devices across everyone is considered. However, the solutions presented in section 3.1 are mostly applied in the American continents.

Concerning the adoption of innovation by companies, in a critical and constructive reflection, it will be considered that companies are obliged to adopt for regulatory reasons in order to ensure transparency between the producer, shipper, regulator, and the end customer. However, we believe that soon, most companies will have adopted blockchain-based solutions.

Scalability on fabric network

Scalability is one of the most relevant factors about private blockchains due to the possibility of choosing the organizations involved. In terms of network configuration, this methodology does not offer significant advantages to growth. Whenever an organization is being added, the network must be paused. Particular attention should be focused on transactions in the network, due to the fact that, if the network goes down, there may be a possibility of corrupted data. With this approach, it is necessary to prioritize a certain number of maximum elements that a network can have.

Scalability on fabric transactions

Regarding network transactions, there is a study Gorenflor et al. (2019) that reports scalability of transactions up to 20,000 per second, since the implementation made for this dissertation is not possible to reach these numbers. On the other hand, in the future, and an environment with more participants, there may be a possibility of high scalability of transactions.

Security and Data Protection

Data security is a crucial factor for guaranteeing reliability. However, the advantages that hyperledger provides since the access permissions and authentication based on what has already been referenced on section 2.11.6, ensures the security and authenticity of data. In terms of Data protection it is frequently discussed subject and has attracted interest in doing quick research. A recent study Finck (2019) reports blockchain equated with European data protection law, with the purpose of contributing to the development of aspects related to the *General Data Protection Regulation (GDPR)*.

Systems integration

The development on system integration involved precise research into software code in such a way that every system can communicate with each other.

It is considered one of the advantages that facilitated communication between databases used in the same server model. However, in counterpart, it was necessary to adapt new network ports to avoid conflicts. The adaptation of the standard BITA aimed to communicate with external entities in order to ensure the integration and interpretation of data.

Continuous development and future work

As future work, it is expected that there will be an increase in the number of stakeholders on the network so that we can perform more operations and transactions. Therefore, it is highlight one of the challenging suggestions for future implementation, namely creating a script to include organizations in the network without corrupting data.

With regard to the system architecture, it is prepared to be highly extensible due to specific modules being created on the various platforms for this operation.

6

CONCLUSION

Continuous monitoring services are a fundamental part of the logistics service, which are used every day to locate vehicles and products. Conventional technologies allow competent monitoring by meeting functional requirements regarding data presentation.

Indeed, intensive research revealed that the logistics services could be improved with the implementation of blockchain, being this technology an added value to ensure reliability, consequently maintain efficiency and effectiveness in response to requests. Furthermore, with the implementation of "distributed ledgers" we can state that all entities involved must have a copy of the data updated whenever there is a transaction on the network, making it impossible to tamper.

In this dissertation, blockchain techniques are used for the following purposes: To use an innovative technology that allows us to guarantee the viability of the data, increasing the confidence of the client, and allowing greater transparency between the production and distribution companies and the client itself. Additionally, we also chose Hyperledger Fabric for the implementation of the blockchain, which, due to the various components and modules, can be adapted for private data authentication and channel mechanism for the organizations involved. Concerning the dispatch of a product, multiple participants are required, and smart contracts are going to make the shipping process more accessible, without unnecessary bureaucracy, and paperwork.

Furthermore, we analyzed the relevant technologies and applications already developed in the market in order to identify business weaknesses. With this, the creation of a global application would benefit several companies from various sectors allowing better efficiency and usability for the user operator and customer. For the development of this platform, it was necessary to embrace and study the requirements engineering. Thus, we created a model that demonstrates the operation of the network and how interactions are going to be made with the final user, not disregarding the adaptation of standards that contributes to the adhesion of companies and the interoperability of systems.

For future work, it is necessary to approach the issue of blockchain scalability, creating new techniques for dynamic network expansion, readjusting the idea not only to one company but to multiple ones, the creation of dynamic channels being necessary in order to guarantee unique transactions for the organizations involved.

In conclusion, our research and implementation determine that the application is prepared to maintain efficiency, ensure product quality and promise to be innovative technology.

BIBLIOGRAPHY

How blockchains could revolutionize the insurance industry (and why they should) . 1 (1):8, 2018. doi: https://assets1.dxc.technology/insurance/downloads/DXC_Industry_Perspective_How_Blockchains_Could_Revolutionize_the_Insurance_Industry.pdf.

Hasib Anwar. Consensus algorithms: The root of the blockchain technology, 2019. https://101blockchains.com/consensus-algorithms-blockchain/, Online accessed 25 June 2019.

Frederik Armknecht, Ghassan O. Karame, Avikarsha Mandal, Franck Youssef, and Erik Zenner. Ripple: Overview and Outlook. 1(1):2,7,8,9, 2015. doi: http://www.ghassankarame.com/ripple.pdf.

Imran Bashir. *Mastering Blockchain*. Packt, 2018. ISBN 978-1-7883-904-4.

Paulo Roberto Bertaglia. *Logística e gerenciamento da cadeia de abastecimento*. Saraiva, 2003. ISBN 978-85-472-0828-8.

BitaStudio WebSite, 2019. URL https://www.bita.studio/.

Leonardo Gondinho Botelho. Um Método para o Planejamento Operacional de Distribuição: aplicação para casos com abastecimento de granéis líquidos. *PUC-RIO*, 1(1), 2003. doi: https://www.marinha.mil.br/spolm/sites/www.marinha.mil.br.spolm/files/arq0063.pdf.

Vitalik Buterin. Ethereum: A Next-Generation Cryptocurrency and Decentralized Application Platform Bitcoin Magazine. 1(1):14–20, 2014. doi: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.

CargoX WebSite, 2019. URL https://cargox.io/.

Greg Carter. An Introduction to Blockchain Technology in Logistics. 1(1):3, 2018. doi: https://teagarden-tech.com/wp-content/uploads/2018/08/An-Introduction-to-Blockchain-Technology-in-Logistics.pdf.

Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance. *MIT*, 1(1):14, 1999. doi: http://www.pmg.lcs.mit.edu/papers/osdi99.pdf.

Qiujun Lan* Chaoqun Ma, Xiaolin Kong and Zhongding Zhou. The privacy protection mechanism of Hyperledger Fabric and its application in supply chain finance. 1(1):9, 2019. doi: https://readthedocs.org/projects/hyperledger-fabric/downloads/pdf/latest/.

Flavien Charlon. Openchain Documentation. 1(1):3, 2017. doi: https://readthedocs.org/projects/openchain/downloads/pdf/latest/.

Chopra and Meindl. *Supply Chain Management, 5th Edition*. Pearson, 2013. ISBN 97801330715o4.

N. Crosby et al. Blockchain Technology. *Sutardja Center for Entreprenership Technology Technical Report*, 1(1):25, 2015. doi: https://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf.

Donald J. Bowersox David J. Clos. *Logística empresarial o processo de integração da cadeia de suprimento*. Editora Atlas, 2001. ISBN 9788522428779.

DHL. Blockchain in logistics. 1(1):28, 2018. doi: https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf.

Mario Dobrovnik, David M. Herold, Elmar Fürst, and Sebastian Kummer. Blockchain for and in Logistics: What to Adopt and Where to Start. 1(1):14, 2018. doi: https://www.researchgate.net/publication/327407561_Blockchain_for_and_in_Logistics_What_to_Adopt_and_Where_to_Start.

Júlio Sérgio Quadros dos Santos, Adilso Nunes de Souza, and Élder Francisco Fontana Bernardi. A Implementação do blochain hyperledger fabric em ambiente linux utilizando containers docker. 1(1):11–16, 2018. doi: https://painel.passofundo.ifsul.edu.br/uploads/arq/201804121724031614655387.pdf.

Edvar, Sasa, Katarina, and Mladen. Blockchain Technology Implementation in Logistics. 1(1):3–0, 2019. doi: https://www.researchgate.net/publication/331325030_Blockchain_Technology_Implementation_in_Logistics.

Elad Elrom. *The Blockchain Developer*. Apress media LLC, 2019. ISBN 978-1-4842-4846-1.

European Commission. Transport in the European Union. *Europe*, 1 (1):176, 2019. doi: https://ec.europa.eu/transport/sites/transport/files/2019-transport-in-the-eu-current-trends-and-issues.pdf.

Eurostat. Freight transport statistics - modal split . *Europe*, 1(1):11, 2019. doi: https://ec.europa.eu/eurostat/statistics-explained/pdfscache/1142.pdf.

Michèle Finck. Blockchain and the General Data Protection Regulation - Can distributed ledgers be squared with European data protection law? 1(1), 2019. doi: https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf.

Go-Ethereum WebSite. Go-Ethereum WebSite, 2019. URL https://github.com/ethereum/go-ethereum.

João Paulo Costa; Joana Matos Dias; Pedro Godinho. *Logística (2.ª edição)*. Imprensa da Universidade de Coimbra, 2007. ISBN 978-989-26-1467-0.

Satheesh Gopi. *Global Positioning System: Principles And Applications*. Tata McGraw-Hill Education, 2005. ISBN 9780070585997.

Christian Gorenflor, Stephen Lee, Lukasz Golab, and S. Keshav. FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second. *cs.DC*, 1(1):9, 2019. doi: https://arxiv.org/pdf/1901.00910.pdf.

GS1. Traceability and Blockchain. 1(1):2–4, 2019a. doi: https://www.gs1.org/sites/default/files/gs1_traceability_and_blockchain_wp.pdf.

GS1. Manual do Utilizador GS1. 1(1):2–4, 2019b. doi: https://www.gs1pt.org/wp-content/uploads/2019/07/Manual-Utilizador-GS1_2019.pdf.

Haber and Sornetta. How to Time-Stamp a Digital Document. *Bellcore*, 1(1):13, 1991. doi: https://www.anf.es/pdf/Haber_Stornetta.pdf.

Niels Hackius and Moritz Petersen. Blockchain in Logistics and Supply Chain: Trick or Treat. 1(1):10–12, 2017. doi: https://tore.tuhh.de/bitstream/11420/1447/1/petersen_hackius_blockchain_in_scm_and_logistics_hicl_2017.pdf.

Denis Domingues Hermida. Sistemas automatic vehicle location e o controle de jornada de trabalho do motorista rodoviário: Mutação Normativa do artigo 62, i da CLT. *Ilos*, 1(1):355, 2012. doi: https://tede2.pucsp.br/bitstream/handle/5912/1/DenisDominguesHermida.pdf.

Parikshit Hooda. Practical byzantine fault tolerance, 2019a. https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft/, Online accessed 25 June 2019.

Parikshit Hooda. Raft consensus algorithm, 2019b. https://www.geeksforgeeks.org/raft-consensus-algorithm/, Online accessed 30 July 2019.

Hyperledger. An Introduction to Hyperledger. *Hyperledger*, 1(1):33, 2017. doi: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.

Hyperledger. hyperledger-fabricdocs Documentation. 1(1):22–25,60–70,92–100, 2019a. doi: https://readthedocs.org/projects/hyperledger-fabric/downloads/pdf/latest/.

Hyperledger. Hyperledger grid documentations, 2019b. https://grid.hyperledger.org/docs/grid/nightly/master/, last accessed on 06.10.2019.

Hyperledger. hyperledger-fabric-msp, 2019c. https://hyperledger-fabric.readthedocs.io/en/latest/_images/membership.diagram.3.png, Online accessed 10 September.

IBM. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchain. *Hyperledger*, 1(1):15, 2018. doi: http://delivery.acm.org/10.1145/3200000/3190538/a30-androulaki.pdf.

IBM. Crypto Anchors. 1(1):12, 2019. doi: https://ieeexplore.ieee.org/document/8645638.

Itweb. Blockchain technology part 2, 2019. https://codeburst.io/blockchain-technology-part-2-smart-contract-fundamentals-d243e2311f94, Online accessed 10 July 2019.

Ben Kothari. BiTAS Tracking Data Framework Profile. 1(1):4, 2019. doi: https://static1.squarespace.com/static/5aa97ac8372b96325bb9ad66/t/5c7e88397817f73e6c60a967/1551796284047/BiTAS+Tracking+Data+Frameowork+Profile+v9_ISTO.pdf.

Timothy Leonard. Blockchain for transportation: Where the future starts. *TMW*, 1(1):57, 2017. doi: http://logisticsandfintech.com/wp-content/uploads/2017/11/TMW-Whitepaper-Blockchain-for-transportation-LaF-Nov-2017.pdf.

Dr. Liew. Hyperledger fabric architecture part 1, 2019. https://www.blockchainguide.biz/hyperledger-fabric-architechture/, last accessed on 12.09.2019.

Jonas Lindholm. How can blockchain transform the logistics industry?, 2019. https://knowledgehub.volvotrucks.com/technology-and-innovation/how-can-blockchain-transform-the-logistics-industry, Online accessed 10 June 2019.

Luiz Ricardo Pinto Samuel Vieira Conceição Lásara Fabrícia Rodrigues, Eduardo Carrara Peixoto. O Problema de roteirização de veículos tratado de forma simples e eficiente em uma empresa transportadora de médio porte. *SPOLM*, 1(1):2–4, 2006. doi: https://www.marinha.mil.br/spolm/sites/www.marinha.mil.br.spolm/files/arq0063.pdf.

Maersk WebSite. Maersk WebSite, 2019. URL https://maersk.com//.

Andrea Margheri and V.Sassone. Using PROV and Blockchain to Achieve Health Data Provenance. 1(1):25, 2018. doi: https://www.researchgate.net/publication/325485559_Using_PROV_and_Blockchain_to_Achieve_Health_Data_Provenance.

Vitor Marques. Utilizando o TMS (Transportation management system) para uma gestão eficaz de transportes. *Ilos*, 1(1):10, 2002. doi: https://www.ilos.com.br/web/utilizando-o-tms-transportation-management-system-para-uma-gestao-eficaz-de-transportes/.

Demiro Massessi. Public vs private blockchain in a nutshell, 2019. https://medium.com/coinmonks/public-vs-private-blockchain-in-a-nutshell-c9fe284fa39f, Online accessed 01 August 2019.

Mic Bowman and Andrea Miele and Michael Steiner and Bruno Vavala . Private Data Objects: an Overview. 1(1):11, 2018. doi: https://www.researchgate.net/publication/326437287_Private_Data_Objects_an_Overview.

Jp Morgan. Quorum Documentation. 1(1):3–5, 2018. doi: https://github.com/jpmorganchase/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf.

Megan Ray Nichols. Will blockchain work in the trucking industry?, 2019. https://www.allthingssupplychain.com/will-blockchain-work-in-the-trucking-industry/, Online accessed 01 June 2019.

Venkatraman Ramakrishna Petr Novotny Dr. Salman A.Baset Anthony O'Dowd Nitin Gaur, Luc Desrosiers. *Hands-On Blockchain with Hyperledger: Building decentralized applications ...* Packt, 2018. ISBN 978-1-78899-452-1.

RoadLaunch WebSite. RoadLaunch WebSite, 2019. URL https://www.roadlaunch.com/.

Jonathan Salama. Blockchain will work in trucking — but only if these three things happen, 2018. https://techcrunch.com/2018/03/02/blockchain-will-work-in-trucking-but-only-if-these-three-things-happen/, last accessed on 01.06.2019.

ShipChain WebSite, 2019. URL https://shipchain.io/.

SkyCell WebSite. SkyCell WebSite, 2019. URL https://skycell.ch/.

Pratik Soni. Location Component Specification. 1(1):13–15, 2019. doi: https://static1.squarespace.com/static/5aa97ac8372b96325bb9ad66/t/5c7e8882f9619a98a55ec24d/1551796355748/BiTAS+Location+Component+Specification+v4.pdf.

KC. Tam. Demo of three-node two-channel setup in hyperledger fabric, 2019. https://medium.com/@kctheservant/demo-of-three-node-two-channel-setup-in-hyperledger-fabric-54ba8a9c461f, last accessed on 19.09.2019.

TEFOOD WebSite, 2019. URL https://www.te-food.com/.

Phuwanai Thummavet. Demystifying hyperledger fabric, 2018. https://www.serial-coder.com/post/demystifying-hyperledger-fabric-fabric-architecture/, Online accessed 12 September 2019.

Sumit V. Hyperledger fabric — part 1 — components and architecture, 2019. https://blog.clairvoyantsoft.com/hyperledger-fabric-components-and-architecture-b874b36c4af5, Online accessed 10 September 2019.

K. Voshmgir et al. Blockchain A Begginers Guide. *BlockchainHub*, 1(1):57, 2017. doi: https://s3.eu-west-2.amazonaws.com/blockchainhub.media/Blockchain+Technology+Handbook.pdf.

Shermin Voshmgir. Smart contracts, 2019. https://blockchainhub.net/smart-contracts/, Online accessed 30 June 2019.

Kinetsu WE. Gps, 2019. https://www.kwe.co.jp/global-en/china-contents/gps, Online accessed 03 December 2018.