Universidade de Aveiro

Ano 2019

Departamento de Eletrónica, Telecomunicações e Informática

ANDRÉ CARDOSO

Sistema para Gestão de Leilões de Dívida com Blockchain

Blockchain Based Debt Auction Management System

Universidade de Aveiro

Ano 2019

Departamento de Eletrónica, Telecomunicações e Informática

ANDRÉ CARDOSO

Sistema para Gestão de Leilões de Dívida com Blockchain

Blockchain Based Debt Auction Management System

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtencao do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientacao cientıfica do Doutor João Paulo Barraca, Professor Auxiliar do Departamento de Eletronica, Telecomunicacoes e Informatica da Universidade de Aveiro, e de Hélder José Rodrigues Gomes, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Àgueda da Universidade de Aveiro.

Dedico este trabalho à minha família pelo incansável apoio e sacrifícios que fizeram para conseguir concluir esta etapa.

o júri

presidente

Prof. Doutor André Zùquete
professor auxiliar da Universidade de Aveiro Universidade do Porto

Prof. Doutor Filipe Correia
professor auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor João Paulo Barraca
professor auxiliar da Universidade de Aveiro

Agradecimentos / acknowledgements

I want to express my special thanks of gratitude to my thesis advisers, Hélder Gomes and João Barraca, for their knowledge, insight and overall support during this year as well to the institution IT UID/EEA/50008/2019. Thanks to them, I fill more mature and thoughtful since the beginning of this project.

To my mother and father, Regina and Artur for their sacrifice, which allowed me to reach my goals. In particular, my brother Carlos that also support more than anyone will know on this final stage. To my youngest brother, David, I hope this journey will serve as an example for his future accomplishments. Furthermore, my grandmother, Irene, my second mother, and her significant contribution in helping me to become who I am.

A special thanks to all my friends that I will not be able to name them all, for their motivation words when I need it most. However, fewer deserve mentioned, Ricardo Fernandes, João Luís, Tiago Fernandes and João Porto to be a person who always believed in me and never finish a conversation without a motivation word to complete this work. Lastly and probably the most important person I met throughout the university, who always share knowledge and dedicated time that helped me finish the master degree, for that thank you very much Rui Lopes.

palavras-chave Blockchain, E-Leilões, P2P Lending, Finanças, Sistemas Distribuidos

resumo No contexto empresarial, quando um comprador adquere um produto, é natural que o seu pagamento não siga de imediato com produto. Assim, identificamos aqui uma falta de liquidez a pequeno ou médio prazo que pode prejudicar um normal funcionamento de uma empresa. Soluções para estas situações existem fornecidas pelos sistema bancário, onde é feito o adientamento, oferecendo liquidez, e por norma conseguem negociar melhores termos. Contudo, nem todas as empresas estão legíveis para receber este adientamento, isto é especialemente verdade quando falamos de Pequenas e Médias Empresas (PME) por causa dos pequenos valores em que operam.
Ultimamente, tem existido um crescimento em plataformas de empréstimos em que os investidores são qualquer tipo de pessoa singular. Peer-to-Peer (P2P) Lending é o nome que se dá a este conceito que visa diversificar oportunidades de investimento através de novas tecnologias e sistemas de informação.
Este trabalho propõe um solução que consiste em dar resposta à necessidade de liquidez imediata das PMEs usando o mesmo modelo de negócios que as plataformas P2P Lending, servindo de mediador entre empresários e qualquer pessoa que queira tornar-se um investidor. A solução proposta é baseada na tecnologia emergente de redes blockchain, a venda de activos por parte das PMEs é de activos de dívida e método de negocio é por leilões.
Apresenta as interações existentes nos processos de leilões e uma solução de software para os suportar.
Identifica algumas possíveis tecnologias e apresenta uma demo desenvolvida juntamente com este trabalho.
Por ùltimo conclui discutindo toda a implementação e as vantagens e desvantagens da aplicação da tecnologia Blockchain num sistema de gestão a leilões.

keywords Blockchain, e-Auctions, P2P Lending, Finance, Distributed Systems

abstract

In the enterprise context, when a buyer acquires a product, naturally, the payment does not follow the acquired product. Thus, we identified the problem of available liquidity at short or medium term, which can arm the business conduct. We may find a solution to this situation offered by the banking system that provides payment advancement, and generally, they can negotiate better terms. However, not only every Small or Medium Enterprise (SME) are legible to be accepted for liquidity advancement. Lately, a new concept as been raise, called Peer-to-Peer (P2P) Lending, in which any person can be an investor. Their goal is to provide alternative investments throughout the new financial technologies and information systems.
This work, suggest one solution that consists of giving to the SME the immediate liquidity desired using the same model than P2P Lending platform by serving as it middleman between the entrepreneur and average person who want to invest. The final solution is based on the emerging blockchain technology, the assets sold by SME are debt-based, and they are sold through auction.
Introduces interactions in auction processes and a software solution to support them.
It identifies some possible technologies and presents a demo developed along with this work.
Finally concludes by discussing the entire implementation and the advantages and disadvantages of applying Blockchain technology in an auction management system.

Contents

List of Figures

List of Acronyms

Fintech	Financial Technologies
B2B	Business to Business
SME	Small and Medium Enterprise
GDP	Gross Domestic Product
EU	Europen Union
GVA	Gross Value Added
FICO	Fair Issac Corporation
DTI	Deb-to-Income
CD	Certificate of Deposit
IRA	Individual Retirement Account
ETH	Ethereum
P2P	Peer-to-Peer
PoW	Proof of Work
PoS	Proof of State
PoA	Proof of authority
PoB	Proof of Burn
API	Application Programm Interface
DTL	Distributed Ledger Technology
SDK	Software Development Kit
CLI	Command Line Interface
GDPR	General Data Protection Regulation
NIFC	Corporate Identity Number
VAT	Value Added Tax
TTP	Trust Third Party
IdP or IDP	Identity Service Provider

1. Introduction

In recent years, more persons are trying to find investments for personal savings. Related to this, several factors are in place, such as the low-interest rate on basic saving accounts and the increased inflation over the last decades. On a more cultural point of view, easy access to information and increased service transparency by the financial system, which led to a 75% grow on investments dedicated to financial technologies (Fintech)[1], from $9 500 million to $22 300 million in 2015[2]. As Patrick Schueffel proposed, after reviewing more than 200 scholarly articles referencing the term Fintech, "Fintech is a new financial industry that applies technology to improve financial activities."[1], which traditionally, has been a time-consuming process with an overwhelming legacy logistics involved, old systems and inert culture within the financial institution, and has constantly been evolving over the years. Fintech it is a concept which has become frequently applied to the process of developing and implementing new technologies to improve and facilitate the process on financial services.

Latter of 2017, we saw enormous media attention to blockchain technology, namely to a specifics, to its financial application introduced by Satoshi Nakamoto in 2008 as "Bitcoin, a Peer-to-Peer Electronic Cash System"[3]. This dissertation does not intend to develop any payment system or do any review to cryptocurrencies and its current financial market. Our motivation decouple the underlying modules and mechanism for understanding how to support other processes, such as auctions.

This dissertation focus on understanding the current context in which an investor is involved in acquirer an asset through current financial services. The focus acquisition's method is focused on auction and the underlying technology involved will be blockchain, a distributed peer to peer system

1.1. Motivation

As consumers in the western world, we are accustomed to seeing fixed prices with small variations related to similar products. However, a product comes from the producer to the final client through a supply chain. Within this supply chain, several interactions take place, especially financial interactions. The negotiation between distributors and dealers is volatile; the price of a product depends on several factors, such as environmental (e.g. crops) and social economics (e.g. local purchase power). Auctions are closely related to commerce operations, and this method is being used through decades to find a balance between the supply and demand. This occurs because it is essential that the producer gets the best price offer to stay in business and different product have different prices for different people in different times and places.

Easily we can find some examples. The Royal Flora Holland[4] warehouse in Aalsmeer, it is one of the largest flower markets in the world where daily are trade around 20 million flowers and decorative plants in the Netherlands. The eBay, which initially starts his online activity as a website for auctions, today it is much more, and a reference for several cases studies through auction literature. Moreover, almost every coastal country has a fish marketplace where auction also takes place.

Today, in the commonly used information systems, we can find security and trust. This occurs because teams of software engineers and business have ethical bonds to his users and regulations. They invest a lot of time, effort and money in maintaining his integrity. The blockchain technology claims to preserve and continues to keep record about whatever information/state the system may deal, with integrity by definition. We intend to explore these properties and try to identify regions where blockchain technology can be applied within our context of auctions.

A large segment that as allowing the development and implementation of new financial technologies is the concept of Peer-to-Peer Lending[5] which is immerging and gain popularity in recent years. They are information technologies (IT) systems that aim the larger possible amount of clients of two types: users looking to invest capital (expecting to receive the invested capital plus interests) and users looking to receive immediate capital (that later will pay that capital plus interest). We will decouple the current processes within this context to understand which types of stakeholder exists and the relations among them for a deeper understanding of the relations in place.

The intentions of this work are also the aggregation of financial vocabulary and literature for self-grow financial maturity for better future financial decisions and share the knowledge it to the readers. Also, the acquisition of literature and practical knowledge of blockchain development and

implementation for future projects. As Daniel Drescher quote on his book "Purely distributed peer to peer system have huge commercial potential as they can replace centralized systems and change whole industries due to disintermediation."[6]

1.2. Objectives

The objective of this work is to present an architecture of a distributed auction taking place with several stakeholders transacting assets among them. It primarily aims to answer the singular:

What are the benefits of applying blockchain technology to auction management systems?

To answer the question, we designed and developed an independent auction system. The context is focused on two types of stakeholders: investors, looking to invest capital and sellers, who owned illiquid assets (e.g. real estate, patents, debt) and are looking to convert them in liquid assets (e.g. cash). The illiquid asset we are focusing on debt titles, owned by a stakeholder that wants to convert them in immediate liquid assets.

1.3. Contributions

The demo solution resultant from this thesis is available into a GitHub public repository which can be found on the link[1].

The repository contains a folder named auctions which is a Java project, Springboot based for the service with relation database. The relation database module is developed with the Workbench base project into the folder name relationaldb. The non-relation database service which is MongoDB framework based, is located into the folder nonrelation-api. Some test_scripts developed throughout this period are located into tet_scripts folder. The client web application is based on the Angular framework can be found in the folder web. Lastly, the Wireshark's capture is located into the folder wireshark-captures. Also available a file, thesis.postman_colletion.json for the Postman application to test some services developed.

[1]Repository source: https://github.com/aCard0s0/thesis

1.4. Thesis Structure

This dissertation is divided into seven chapters, described by the following:

Chapter 1 has a brief introduction to the topics addressed in this document. Also, an explanation of the context, motivation for this solution followed by the objectives.

At Chapter 2, we present and contextualize several subjects we mentioned along with this document. It starts with an introduction about some financial concepts following by definition of the several types of stakeholder presented in our context. Also, it gives an introduction about a new trend, Peer-to-Peer Lending and how the market and users are responding to it. Lastly, we introduce auction theory and e-auctions solutions.

Chapter 3 is entirely dedicated to blockchain technologies, where we explain the environment of the application, the problems it solved, and what component are presented. Also, we mentioned some frameworks presented in the market to explorer the blockchain capabilities.

Chapter 4 presents a complete picture of the problem. It defines the principles and introduces the solution this document defends throughout the rest of the work.

Chapter 5 shows how a demo solution was implemented based on the principles define in Chapter 4. It presents a practical approach to the problem.

Chapter 6 presents results and discussion from the solution design and implementation process mentioned in Chapter 5.

Lastly, Chapter 7 is given an overview of all process of developing this thesis.

2. Background and Related Work

Although the work is focused on engineer processes and mechanism, we will frame the financial context. We will start by introducing some basic financial vocabulary needed throughout this literature. As also identify the stakeholder in place, its definitions, relationships and interactions, often comparing the traditional process with more technologic and trending alternatives.

2.1. Introductory Financial Concepts

This section presents the current concepts of financial literacy. The definition enters in some details and is based on the literature that can be found mainly in the website *investopedia*[7], an online encyclopaedia about financial terms. Some other terms, less used on this dissertation, may not be defined, but we will present references to it. The defined terms are the following:

Asset

A possible definition is presented as "An asset is a resource with economic value that an individual, corporation or country owns or controls with the expectation that it will provide a future benefit." [8] Basically, a resource that can generate cash flow, reduce expenses or improve sales. It can be divided into categories for the purposes of taxation or to measure the financial health or the value of an entity. The following presents significant categories.

- **Intangible Assets:** are assets that are difficult to determine the value, typically do not have a physical presence. For example, copyrights, trademarks, patents. In contracts, tangible assets have a physical representation.
- **Current Assets:** are assets that are expected to be sold, consumed or a service to be converted to cash on the current operating year. It is a short-term economic resource that takes the form of inventory, prepaid expenses or cash.
- **Fixed Assets:** are long-lived or permanent assets that cannot be easily converted into immediate liquidity, e.g. money. Such an example can be buildings, pieces of equipment or even a patent. Typically, in case of equipment, with the ageing and use, it depreciates according to with proportion to its useful life. While in the case of properties, the trends and economic growth in the given region have an important factor in the price.

- **Financial Assets:** it is an asset that has a value that is based on a contract, e.g. securities, bonds, stocks.

Debt

It is essentially, an amount of money borrowed by one party from another with the condition that will be paid back at a later date. It takes several forms: for an individual, it is a credit card, auto loans or mortgages: for business or corporation, it has more options such as bonds and commercial paper[9]. For both enables the short-term purchase power that could not have a place under normal circumstances.

Loan and loan notes

A loan may be done with any asset, more common money, it is "given to some party in exchange for future full repayment along with interest or other finance charges "[10]. The terms and conditions are agreed before the loan occurs. Typically, two types of loans can be made, secured or unsecured, which mean the borrow assets may be backed by collateral or not, respectively. By collateral, we mean by another asset that may be used to pay out the loan.

This promissory loan agreement generates a loan note[11]. That essentially contains detailed information about the loan, such as names, contact information, principal lend, interest rate, payment scheduler and due date. Additional information regarding terms and conditions may include repercussions or penalties for late payments. It is considered legal, valid agreement until the borrower pays the amount listed.

Interest rate

According to [12], "Interest rate is the amount charged, expressed as a percentage of the principal, by a lender to a borrower for the use of assets." Typically, charged at the end of the month, quarter or year. It is the cost of the debt for the borrower and the rate of return for the lender. The rate ranges according to the risk of the lender be paid back. Two types of interest can be charged, presented below.

Equation 1, represents the simple interest rate, to compensate the lender for the loss of use of the money during the lending period.

$$Simple\ Interest = Principal \times Interest\ Rate \times Time$$

Equation 1 - Simple interest calculation.

Equation 2 represents the compound interest. It is not only applied to the principal but also the interest accumulated in previous periods. For instance, the borrower owes the principal plus interest for the initial period; on the second period, the interest rate takes into account the principal plus the interest rate from the previous period. Takes the formula:

$$Compound\ Interest = Principal \times [(1 + Interest\ Rate)^{n} - 1]$$

Equation 2 - Compound interest calculation.

When entities save money, compound interest is more favourable for allowing the banks using the funds. For example, to compensate the entity, the bank can pay up to 6% interest into to account annually, while the bank is taking 15% from the borrower. Let the bank netting 9% in interest.

Credit

Credit[13] can take different formats, depending on the context in which it is inserted. It can take the form of contractual agreements, where banks offer lines of credit for general goods, car loans or mortgages. Basically, individuals or business can make large purchases without paying the entire price upfront.

Over a period agreed between the parties, the borrower repays the loan, plus interest, until the credit is fully repaid. This case increases consumerism, thereby increases sales for retail or business to business (B2B) sellers. Another example occurs between supplier and distributor. The supplier gives products or services to the distributor and does not require immediate payment until, normally, at the end of the month or end of the quarter. In the case of a bidirectional relation, financial settlements between the identities are done periodically.

Defaults

Occur when an entity cannot fulfil the original agreement to pay a loan, debt or credit. When such an event happens, usually a legal action is involved. If the risk was not well-considered, it

appears as bankruptcy for a business, a sovereign or national default for a country, and for an individual, he may lose the good acquired with the money from the debt or credit.

Liquidity

"Liquidity refers to the availability of cash or cash equivalents to meet short-term operation needs."[14] Cash is considered the standard for liquidity. If an individual or corporation/business needs to buy a product or service in a short-term notice, the assets that can easily be used to obtain it is cash. If it is not available, but owned an asset with similar value, it will have to sell it in order to use the cash to purchase the product or service needed. This may be fine if they are willing to wait a few weeks or even months, but for immediate purchase, this can be an inconvenience to have liquidity desire. As a result, the seller may present the assets with discount, instead of waiting for a buyer willing to pay the full value.

Liquidity ratio, also called the current ratio is the company's ability to pay short-term obligations.[15] Typically used as an indicator of how the business is performing. Equation 3 describes the percentage of liquidity own by an entity in simple terms.

$$Current\ Ratio = \frac{Current\ Assets}{Current\ Liabilities}$$

Equation 3 - Current ratio calculation.

As an investor, a company that has poor liquidity, usually have more difficulties in growing and increasing performance because short-term funding isn't available. May also indicates that exists inefficiency in generating revenues with its assets. In the other hand, high liquidity may indicate the company isn't investing wisely its resources to grow operations. In conclusion, it is of high importance for a company to have liquidity for reacting to unforeseen events without compromise operations.

Financial Instrument

"Increasingly, financial instruments that have been standardized are stored in an electronic book-entry system as a record, and the parties to the contract are also recorded."[16] It is a general term for a collection of monetary assets widely used in the modern economy. Although they have been standardized, many are custom agreements that the involved entities tailor to their own needs.

For instance, a financial instrument may represent an object that is used to exchange money for:

- Future repayment of principal and interest, which is the case of Bonds.
- Possible capital gains or interest such as stocks and funds;
- Possible capital gains or to offset risk, Currency, Swaps;
- Protection against risk through insurance;

Typically, this term is classified into types. If it is cash or derivate instruments. Also, it can be classified into the asset class, for example, if it is debt or equity-based.

Small and Medium Enterprise (SME)

Although it is problematic to estimate exact numbers, this report [17] by Edinburgh Group, tries to aggregate several surveys and studies to clarify the impact of SME into the global economy. This report suggests that more than 95% of the enterprises across the world are SMEs, with significant contributions to low-income countries, both to Gross Domestic Product (GDP) and employment. SME's also has become a major contributor to innovation through collaboration with larger corporations, become embedded in the supply chains of larger businesses.

On the same report, it is estimated for 27 countries in the European Union (EU-27), that "They (SME's) account for 99.8% of all enterprises, employ 67% of all workers and contribute 58% of Gross Value Added (GVA)".

2.2. Lending

Banks have been the leading organizations where people get capital for numerous applications. They are the traditional lender mainly to SME's and individuals through credit lines. Figure 1 presents the traditional banking system, which is responsible for an aggregate, manager or provides funds to clients: depositors and investors.

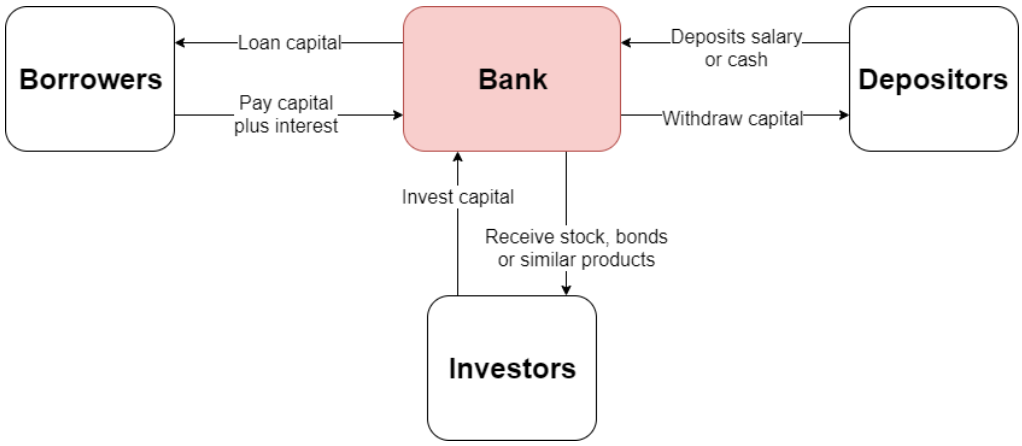

Figure 1 – Traditional banks interaction.

The borrowers are the bank clients that apply for the bank to lend capital to invest in some personal business or general goods consummation. The depositors are the clients that use the bank services as a trusted authority to deposit their salary or other savings. These regular clients are encouraged to make several deposits over a period in order to receive interest over it. Terms and conditions are diverse and depend on the chosen bank product, the government laws and the depositor negotiation power. Overall, the interest rates the depositor receives are low as well as the risk that is taken. The investors are another source of liquidity used to increase cash flow for more applications. Typically, they are the clients who want to receive higher interest rates over his capital invested and assume higher risk as well. Note that one individual may interact with the bank as one or more types of clients presented.

Although we will continually refer to the bank as the primary source of loan and credit through this section, It is important to note that they are not the only type of financial institution with the ability to hold or provide capital. However, in this section, we focus on traditional institution examples and procedures once it was the origin of this type of service.

2.2.1. Banks

When analysing the lending viability, the payback modality is probably the most crucial factor to make the final decision. It specifies the terms of how the loan or credit are made, in which period is paid back and what are the terms and consequences of default. To evaluate the risk, the financial institution may rely on the applicant's credit history, for example, Fair Isaac Corporation (FICO) Score[18] or Deb-to-Income Ratio (DTI)[19]. Theses credit history consists of a record of credit taken, as well as amounts owed and payments status. FICO score aggregates all this information and more to predict consumer behaviour and to sell to other companies as an indicator. Often done by a third party, it is considered a standard in the industry. Among other services, "FICO also maintains a fraud protection service used to safeguard more than 2.5 billion credit cards."[20] The DTI is a ratio indicator, in which a low ratio demonstrates a right balance between debt and income, thereby enhances the change to get approved for a loan.

These evaluation indicators and methods may or may not be used together. Extensive search also requires more work hours, thereby costs, which is reflected in the interest rates for the borrower or cause the profit margin to decrease for the bank. If they are not good enough to acquire the requested loan, it is considering unsecured[21]. Typically, it is denied, or the interest rate will be higher than average. A solution for the borrower in such cases can be the use of assets as collateral, called lien[22], which give the lender legal right, granted by the owner, to seize the asset if the obligation is not satisfied. This way, the borrower can ensure the requested loan to be approved or even decrease interest rates.

2.2.2. Borrowers

According to [23], "Borrower perceptions of loan burdens are important. They provide insight into the willingness of consumers to use the product and to recommend the product to others." Someone who is considering obtaining a loan to start a business or credit to buy general consumption goods, invests a significant amount of time trying to anticipate the burdening effect on his income. Using debt may be a double-edged sword, it may help achieve financial goals or compromise it. If it is good or bad debt, highly depends on the context it will be used and what term and condition are into it.

Usually, a mortgage to buy a house is considered good debt. Like student loan, they have low-interest rate due to the time the person typically takes to pay back the full amount of lend money. However, houses have a market that is not stable. Which mean they can increase or decrease in value. In an ideal situation, a house increases in market value, thereby the interest paid over time can be annulled. However, the inverse situation is equally possible. Depending on the conditions agreed to take a mortgage, it is possible that the financial institution that lends the money, take the house if the borrower ability to make the payments is compromised. Such was the case of mortgage-backed securities in the financial crises 2008[24].

Payday loans are cash advance services, a short-term unsecured solution to borrowing a small amount of money by a non-traditional financial service provider. The lender charges high levels of interest and charge fees that the borrower is not aware of. In the case study[25] conducted by the Financial Consumer Agency of Canada[26], we can find Chart 1, that shows the cost of borrowing a 300$ loan for 14 days.

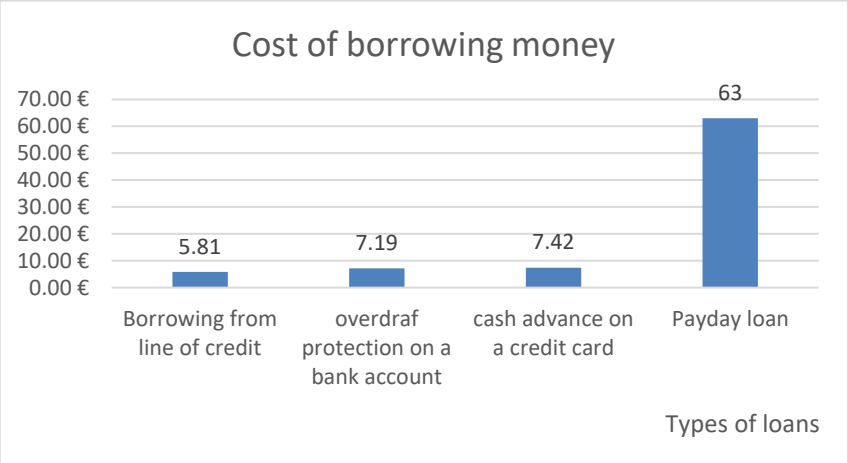

Chart 1 - Payday loan cost vs other ways of borrowing money.

Payday loans are considered an expensive way to borrow money. As well stated in the same study "that the majority of respondents were not aware of the relative costs of all short-term credit options and may be using payday loans more often as a result." This type of loan has a bad reputation, with term and conditions, often misleading the customer with ambiguous language. Such was a scandal occur in the US were "From 2007 to 2013, Hallinan sought to collect more than $690 million of illegal debt and successfully collected $492 million, prosecutors said."[27] Fortunately, in recent years, a regulation was put in place to prevent related issues, such as guidelines for a technical guide for microfinance institutions, present in Commercial Loan Agreements[28].

As an entrepreneur to create a start-up business, it is not uncommon to apply for a loan. A company that uses large amounts of debt may have difficulties to pay out the interest rate if the revenue stream drops for some reason. However, a company that does not use debt may be missing opportunities to expand the operation. Different industries use debt differently, it highly depends on the context and type of business they operate. Thereby do not exist a "right" amount of debt that must be used.

2.2.3. Investors

Investing through a bank, it highly depends on the context that better suit the goals. First, a distinction must be made between two types of investors. Figure 1 shows two entities call depositor and investors. The main difference to notice is that a depositor is a regular person who deposits a family budget in order to run their daily routine. On the contrary, an investor is a person who got an education in some sort of financial subjects or at least has more interest in applying his funds on higher risk financial instruments.

For the depositor, once he is a bank client, several options exist to consider, each with own time requirement and characteristic:

- **Basic saving account**: it is the standard and most straightforward way to invest. The interest rate is low, but the funds are available in case of unforeseen events, usually with small penalties or fees. The bank will lend the money at a higher interest rate to other persons in order to cover the interest rates paid to the depositor.
- **Certificate of deposit (CD):** it is basically a closed saving account with the interest rate significantly higher. After investing, it cannot be accessed without a significant substantial penalty than the basic saving account. There is a mid or long-term strategy option (5 up to 30 years). However, after taxation and inflation in recent years, the account value is almost the same. This option has been losing interest among investors.
- **Individual Retirement Account (IRA):** it is a mid-term strategy of investing while avoiding taxes. The bank has a small group of people responsible for applying the capital in several mutual funds, bonds or stocks. For example, stock such as S&P500, an index that contains the value of the higher 500 companies or PSI20, a similar index that contains the value of the higher 20 companies in Portugal.

The option to invest in the stock market without being through the IRA framework is possible having a private stock manager or even on your own. However, technical analyses and financial culture are needed in order to obtain success. That is the knowledge that professional investors learned from years of experience.

The investor participates in a parallel market, which can sell and buy shares to each other. Some financial instruments owned by the bank, typically fund,s when sold to this type of investors, allows them to lower their exposure to the risk and at the same time increase liquidity. The investor takes a position, earning a rate return associated with the payments from the debtors. Even though tangible assets back the funds, there is no guarantee that the assets maintain or increase in value.

2.3. Peer-to-Peer Lending

Within the context of Fintech, a service is provided by making use of modern technologies and the online environment, the Peer-to-Peer (P2P)-Lending services. These services have begun having adherence since the 2008 financial crisis because they cut out the middleman and the borrowers and clients may find better terms and conditions than their local bank.

Services offered by a P2P Lending platform can be simplified and automated by a fraction of the average cost. This saving is possible because it dematerializes the banks and all the cost associated with it. Figure 2 presents a general architecture of these services, based on the research report "The Rise of the new Shadow Bank"[29] by Goldman Sachs[30]. Once in a technology environment, the involved entities had no borders, thereby an investor or consumer can find capital at excellent rate returns in any part of the world. Also, monitoring can be done in more detail hence better-customized service.

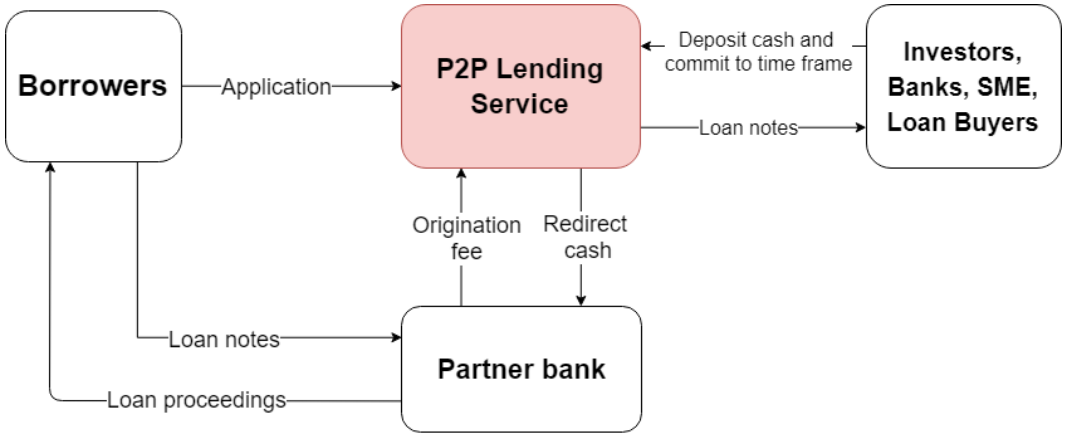

Figure 2 - P2P Lending interactions.

P2P lending differs from the traditional lending system mainly to the fact that the money borrowed is not from the platform but from the investors, thus offering services to consumers and also to anyone who wants to become an investor. It offers convenience for communication and payments through the online environment. Also, it may offer more control and transparency over funds and investments. Investors can manage the assets in which it is investing, and consumers may find more simplified access to credit. Entrepreneurs can also propose business plans and find investors willing to invest the capital to start the business. In short, it brings borrowers and investors together on the same platform.

Typically, P2P lending services have partnerships with financial institutions, banks, not only to facilitate access to capital but also for regulation purposes. These partnerships have mutual benefits because the bank does not have to concern with the development and maintenance of the online service while reaching more clients. In return, the P2P platforms get fees for each payment processed, or client acquired.

2.3.1. P2P Lending Versus Banks

By extending the range of people and presenting new ways of applying to advance capital, it raises the question if this new type of banking service is a substitute or complement to the traditional banking system. Trying to answer this question, we analysed a paper "Peer-to-Peer Lender versus Banks: Substitutes or Complements?"[31] by Huan Tang, where it developed a framework in which P2P platforms can operate as substitutes or complements to banks. The data used for this paper, was "detailed information on all loan applications and funded loans between 2009 and 2012", from the platform LendingClub. Details about the operation of this platform can be found in Appendix B.

The paper provides insights on the relation between P2P platforms and banks by considering three cases, which are: P2P platforms are perfect substitutes, perfect complements or an intermediate case. The predictions made are categorized into three groups: volume, distribution and frequency. Starting with volume states that, "the results on P2P volume show that when banks cut lending in the consumer credit market, borrowers switch from traditional financial institutions to P2P platforms." This means that using this indicator on its own, we cannot conclude if the P2P platform is complements or substitutes. Analysing the distribution, "P2P borrower quality is thus consistent with banks and P2P platforms being substitutes". Lastly looking to frequency "P2P platforms experience an increase in originations among low-quality borrowers, in line with P2P platforms being substitutes to banks."

It is important to notice that these results may not generalize to other markets, such as the residential lending market. A limitation of the above referred paper is that it is focused on the banking market of the U.S., which may not apply to the banking market structure of other countries. Furthermore, the Fintech industry is changing rapidly. Taking this into account, it concludes that "P2P platforms may not operate as substitutes to banks in the long run." And it is stated "P2P platforms complement banks by focusing on the market segment for small loans. The amount requested by borrowers migrating from banks to P2P platforms is larger than 90% of pre-existing P2P loans."

The conclusions of that paper help to understand the types of services and clients that work better with P2P Lending and with traditional banking service. One limitation that we find for our context is that it does not consider SME on accessing advanced capital.

2.3.2. Conclusions

Investors are always seeking new ways to diversify their portfolios by investing in alternative assets. Compare to current rates for a savings account that ranges from 0.1% in Belgium to 2% Slovakia after taxes, according to to the banks.eu[32] Website. However, the deposits in banks at the European Union are protected up to 100 000€, in case of bank failures. In case of the defaults of a loan, in most cases, the losses are borne by the borrower.

Markets work in cycles, assuming some decline in the local or global economies will happen. As a consequence, when people start losing a job or economic power, the first thing that stops paying is obligation debt, such as credit card debts and personal loans. Furthermore, such times are a great opportunity to buy to take advantage of low prices. However, it may be difficult because the money is tied up on these P2P platforms.

Theses platforms hold great potentials by using technologies for more connectivity between people over the world. The ability to accept a potential borrower or investor into the platform goes beyond the credit rating agencies and financial statements. The information obtained through an online form, documentation they provide, social media and demographics, it promotes convenience, efficient and automated process by algorithms instead to handwritten process, thus reducing a large number of operational expenses. Also, it is capable of being more transparent and trustworthy to all entities involved if the right technologies are implemented. Nevertheless, it is a relatively new concept and not very well-known, compared to the traditional economic system it still is in the early stage and needs maturating in the global economy.

2.4. Auctions

A definition for auctions can be “a market institution with an explicit set of rules determining allocation and prices on the basis of bids from market participants”.[13] Basically, an auction is a mechanism where several buyers are competing to buy an item from a single or multiple sellers. It can take several forms with minor variations within the procedures. We will present five basic types: English auctions, Dutch auctions, Frist or Second Price Sealed Bid and reverse auctions. Following we characterize each one:

- **English auctions:** considered a simple traditional auction, the auctioneer starts by putting a low price in the item. Then some bidder proposes an offer price higher than the current best bid price. The auctioneer stops when as no more bidder to outbid the current best offer.
- **Dutch auctions:** also called multi-unit auctions. The auctioneer starts the auction by putting a high price on the item. High enough to be sure that nobody wants to bid the item. Then the price gradually decreases until one bidder decides to buy.
- **Reverse auction:** into this type of auction, it is the seller who bids the price that is willing to receive for his goods or services. It is used when there are several sellers to one buyer. The seller who is willing to sell this product for lower price wins.
- **First Price Sealed Bid:** each bidder proposes a sealed offer to the auctioneer. Once all offers are made, the auctioneer ranks them from the highest to the lowest one. The bidder who proposed the highest value wins the item.
- **Second Price Sealed Bid:** also called Vickrey auction, it is very similar to the First Price Sealed Bid. The bidder who wins the auction, as before, is the one who offers the highest value. However, he only must pay the amount offered by the second-highest bid.

Note that here we find two categories. The first three are open auctions, while the last two are sealed-bid auctions. Meaning the English and Dutch auctions require the bidders to be present in the same place; hence a bidder may observe the behaviour of the competitors, while the bid in sealed-bid auctions may be submitted by letters or email.

Also, we can observe two situations: when there is one item to sell and several buyers, we may use any type of auction mentioned before except the reverse auction; when only one item to buy and several sellers in competition, we may use a reverse auction.

The auction theory literature is vast and a subject of discussion for several decades. A lot of terminologies exist. Below we present the more commonly used.

- **Reserve Price**: a price set before the start of the auction. When it is close, if the final price is lower than the reserve price, the seller has the right not to sell the item.
- **Bid increments**: the minimum acceptable price difference between two successive accepted bids.
- **Buyout price**: A price defined before the start of the auction that is the acceptable value to sell. If a bid is greater or equal than the buyout price, the auction automatically stops, and the item is sold to the bidder.
- **Multi-round or one-off auction**: One-off auction as the name suggest, is a typical one-time auction. However, things can be more complicated, auctions may be multi-round used for bigger corporations to emit licenses, for example. The bidding is set through several rounds, it stops when there is a round with new bids. This may take weeks or even months.

Participants attending an auction may try to manipulate the outcome of an auction, on any side, buyer or seller. Regardless of the violation origin, it harms everybody. Some problems know are:

- **Snipping***:* bidding the item on last possible second of an auction;
- **Bid shielding:** a process that is illegal by law. Consist of two bidders working together, where one made a bid high enough to discourage other potential buyers from bidding. Thus, at the last minute retract the bid leaving the item being sold a lower price to his partner.
- **Shill bidding:** this practice consists of placing a bid on an item to inflate the final value.

Other bad behaviours that may occur is the seller be contacted by a potential buyer to cancel the auction and sell the item through a private and normal transaction of goods. To prevent this behaviour, typically, auctions may present a reserve price, that allows a potential buyer to acquires the item by paying the full amount set as the reserve price.

Some solutions for these problems may be found in the literature dedicated to auctions. However, we will not focus on such solutions in an off-line environment.

2.4.1. e-Auctions Solutions

With the development of e-Commerce, we might think that a natural adaptation occurred applying auctions to sell and buy goods for the general public on the internet. But according to to the paper "Auctions on the Internet: What's Being Auctioned and How?"[33], the auction phenomenon appeared on the internet much early. The author observed, "Before NCSA Mosaic (the first Web browser for the Windows and Macintosh platforms) was released at the end of 1993, there were already a number of auctions taking place on text-based Internet newsgroups and email discussion lists." The newsgroup was devoted to the trading of collectable trading cards for the game Magic: the Gathering.

Accordingly to the same paper, the earliest web-based auctions (e.g. e-auction) appear in 1995 on two platforms, Onsale and eBay. From there to recent times, several other e-auction immerge like Amazon liquidation auctions[34], and some specialized in specific king of goods such is the case of Royal Flora Holland[4], auction marketplaces for flowers in the Netherlands and Docapesca[35] for fresh fish in Portugal.

The cost of doing business with auctions is not cheap, as the commission paid by the seller and buyer may vary but is not uncommon to find 10% and 15% commission for both sides. For e-auction, however, the paradigm changes. Due to the natural environment, several advantages appear such as geographic independence, larger public audience and time flexibility. There is no necessity of physical space to attend the auction and all logistics, and personnel costs can be cut, leaving higher-margin available. For instance, eBay, depending on the good to be transacted, may charge less than 2 dollars to start the auction and when sold the commission may be less than 5% of the final price.

2.4.2. e-Auction deviation behaviours

Although the several advantages the e-auctions are not exempt from problems. The problems identified in Section 2.4, snipping, bid shielding and shill bidding, appear now on the online environment. These behaviours may be easier to notice but are more frequent. Also, new types of violations appear, such as transaction interference and transaction interception, concepts introduced by eBay online auctions.[36]

E-auctions online platforms implemented mechanism and rules to prevent these violations. For example, the snipping problem, if detected higher activity during the last minutes of an auction, the length of that auction may be extended for a specific delay.

For instance, eBay resolved the problem of bid shielding by using a restricted policy to cancel a bid made by a potential buyer. Customer service states that it is only possible to cancel a bid if the time to close the auction is superior of 12 hours and one of the following situations occurs: It is not possible to contact the vendor; the description of the item was substantively modified after the bid; or it was typed the wrong number, for instance, the potential buyer typed 500 euros instead of 50.

The shill bidding problem may be fairly easy recognizable by checking a potential buyer bid history. Someone who is constantly bidding on items and never wins may be suspicious. Also, in almost every platform, it is common to appear highlighted sections with the auctions that are trending at the moment. Among various variables, one method that may be used to set if an auction is trending is the number of bids that it is receiving. A seller may bid with multiple "rogue accounts", to promote his auction.

The behaviour of transaction interference occurs when a potential buyer is bidding on to an auction item and receive an email from another seller offering the same item for a lower price. It is not a situation that can be easily noticed and prevented. The solution present by eBay is advice to ignore the message and leave negative feedback to the seller who sends the email.

The transaction interception occurs when a person is keeping track of an auction about to be closed and then, e-mails the winner as if he was the seller. Normally, the email looks very genuine. It treats the winner very politely and asks for the payment. It has the same characteristics as phishing websites. It is an illegal behaviour because it is stealing from the auction winner. To be protected against such attack, it is advised to use some protected payment service.

2.4.3. e-Auction and Blockchain

Some organizations have been developing e-auctions projects and using blockchain. Bellow, we analyse a project called Auctionity[37] created by DomRaider Group[38] that are taking advantage of auctions alongside with the underlying integrity of the blockchain, providing real-time auctions based on Ethereum Network using Ethereum[39] coins (ETH).

They start by defining several entities such as the *Seller*, *Bidder* and *Winner*. Also introduces new entities such as *AuctionReferrer*, the entity which allowed creating an auction and the

BidderReferrer, the entity which allowed the bidder to place a bid. Also, introduce the concept *AuctionityDeposit* which is an escrow service created by Auctionity for the Ethereum Live Network. Lastly, the *Auction* which is a smart contract created by the *Seller*.

They also explained how their smart contracts work in a typical auction on their platform. The seller transfers the ownership of the asset (e.g. token) to *AuctionityDeposit*, which will provide several guarantees. That guarantees are: the asset will only be transferred to the winning bid if the seller signed the auction creation; the auction can only be created for an asset by his seller; And on auction conclusion; the seller will be paid, and the buyer will pay; if there are no bids, the asset will be returned to the *Seller*. These rules are written in smart contracts. And the *Seller* after transferring the asset to *AuctionityDeposit* can only set some parameters such as the auction start amount, start date or bid increments.

Figure 3 is an image presented into Auctionity documentation where then show some features of the project. As we can observe, the token is transferred from the Ethereum Live Network to the private ledger they own where it will execute the pre-defined contracts.

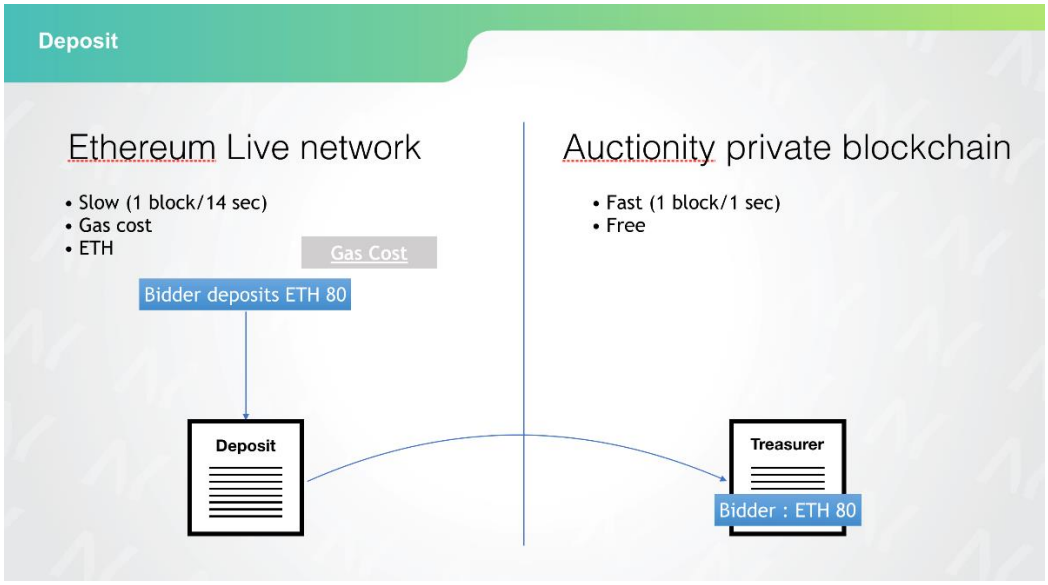

Figure 3 - Auctionity deposit scheme[1].

To bidder must own the total amount needed on *AuctionityDeposit* to bid into any auction. Again, providing guarantees such the ETH will only be used by the *Bidder* who deposited it, he can

[1]Image source: https://docs.auctionity.com/docs/deposit-eth-on-auctionity

withdraw as long is not the current highest bid and the ETH will be transferred to *Seller* and the asset to the winner if the bid is signed.

When a bidder transfer occurs on Ethereum Live Net to *AuctionityDeposit*, this deposit occurs in the Main Network, and the Oracle replicates on the Auctionity Sidechain, acting as a mirror. But the user must pay fees as a normal transaction because Auctionity uses private blockchain, that allows them to accelerate the block creation for real-time use. While new block into Ethereum Main Network is created every 14 seconds, the private Auctionity Sidechain allow a block creation every second. Thus, if bidder loses the lead on an auction, it can try outbidding his competition, withdraw or bid another asset instantly.

The withdraw process appears to be simple as long the amount required is consistent with the available amount. The platform emits a withdraw voucher, and the user had to pay fees again because the process does not occur into Auctionity Sidechain, instead, occur Ethereum Live Network. Note that the fees are only paid when sending and withdraw ETH to the Auctionity platform. Within the platform, every bid is free.

3. Blockchain Technologies

On this chapter, the focus is trying to understand the underlying mechanism and environment of the blockchain technologies. We will present different types of architecture, modules and the resulting properties. Also, we will briefly introduce to several consensus algorithms, existed types of blockchain and some frameworks

A definition of blockchain can be, “is a shared, trusted, public ledger of transactions, that everyone can inspect but which no single user controls.”[40] From that quote, we find two properties that this technology presents, transparency and integrity. Although, it also says that no single user controls, this statement depends on the type of blockchain been used as we will discuss later in Section 3.5. The blockchain technology offers solutions by presenting data with integrity and transparency. Furthermore, every alteration made on the ledger is done by a signed transaction which is linked to the authenticated user, and every transaction that occurs into the ledger is immutable. Thus, authentication and immutability are functional aspects.

3.1. Blockchain Operation Environment

Currently, we find three types of environment for any application: centralize, decentralize and distributed environments. Independently the goal set for any application, any of these architectures can be used to achieve it. To develop a system, the architecture used, it is just a means to an end. However, each one has advantages and disadvantages, and different methods to achieve the same result.

Blockchain has a distributed architecture often called Peer-to-Peer system, that is composed by several nodes, e.i. peers. Every node has the same features and responsibilities, and we never know if all nodes are trustable or if any has malicious intentions. Thereby integrity and the ability to identify ownership, by the application context, is crucial to have success.

Figure 4 illustrates the three different types of architecture mentioned. From right to left we have a centralized, decentralize and distributed computer system respectively with node represented as dots. Where the orange node only acts as a server to his peers which are the connected black dots. The blue nodes also act as a server to his peers but may act as a client between similar computer systems. The green nodes may act as a server or client to any other node within the network.

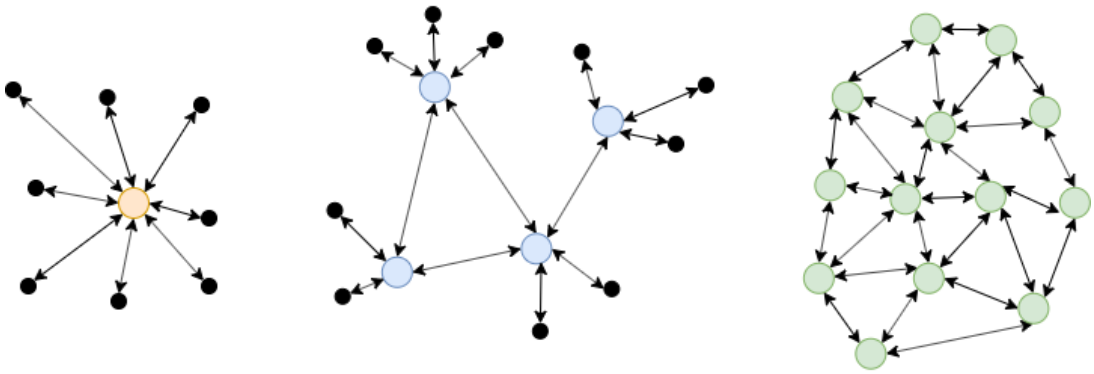

Figure 4 –Centralize, decentralize and distributed architectures.

Any architecture represented is capable of sharing resources such as storage, processing power, network bandwidth and data between his peers. The centralized application, the right figure has a central computer system that controls every move. Easily, we can identify problems in scalability, high overheads and single points of failure.

The decentralized application is a hybrid solution. It can attenuate these problems and eliminate the problem of a single point of failure. However, it adds more complexity because all central system must be synchronized first between them only then it can synchronize to their peers.

The purely distributed system each peer is equal. Thus the system gets stronger when added more peers. Nevertheless, it comes with some pitfalls, “Dishonest and malicious peers comprise the most severe threat to the peer-to-peer system because they attack the foundation on which any peer-to-peer system is built: trust.”[6]

The use of blockchain it highly depends on the application purpose. For instance, it is possible to use blockchain on a centralized system. However, we must ash: If it is needed to verify the integrity or reach consensus for the data produced by itself. Probably it is not needed. To the decentralized system depends on the context. Following the representation in Figure 4, if one or more blue node is from distinct entities, to create integrity or consensus, blockchain may be the right solution. Otherwise, if they belong to the same entity, it may not be justified it used. To a distributed system, as stated in the book “Blockchain basics: A non-technical introduction in 25 steps”. “The core problem to be solved by the Blockchain is achieving and maintaining integrity in a purely distributed peer-to-peer system that consists of an unknown number of peers with unknown reliability and trustworthiness.”[6]

3.2. Underlying Mechanism

Blockchain operates in P2P networks, and also uses two more concepts: cryptography and may integrate game theory, depending on the context. Cryptographic mechanisms ensure the data is not tampered and is used to authenticate participant; Game theory acts as an economic incentive to promote peer's participation and ethical behaviour.

The problem solved is formally known as the double-spending problem[41], which consist of duplicate information to be used in two distinct situations. Typically, there is no problem in duplicate information. However, in the economic context, it is a severe problem. It promotes inflation, and an asset that can be duplicated at will loses value. The following will introduce how particular mechanism operates in order the solve this problem.

As the name suggests, blockchain is a chain of blocks. Each block is identified by its cryptographic hash which is a unique value with fix length. Also, it contains a ledger of transactions which is a list of every data transfer between users, the hash from the previous block, and the timestamp of the block creation. Any node in the network has a copy of this chain, thus knowing the current state of the data that is being exchanged Figure 5 represents the chain of blocks, and it is content.

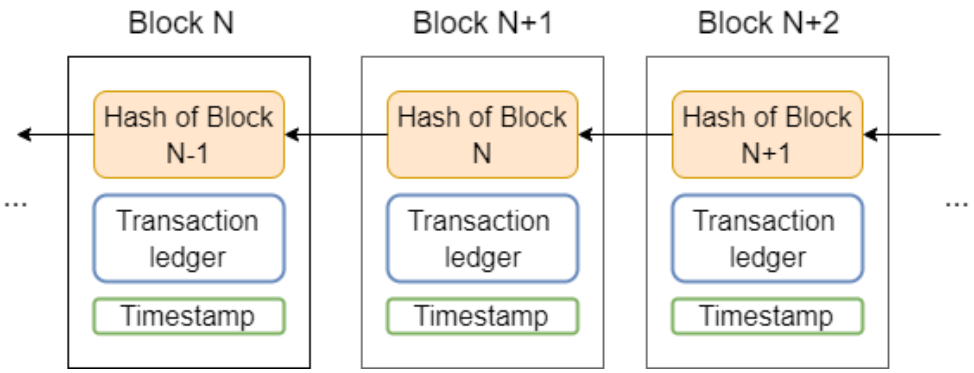

Figure 5 - Representation of a chain of blocks.

The user interacts with the blockchain using asymmetric cryptography[42], which is based on a key pair with a private and public key, and provide two cryptographic properties: confidentiality and authentication. For instance, to send data with confidentiality, we first fetch the destination's public key, encrypt the data and only using the destination's private key, one can decrypt the data. To send

authenticated data, we first sign the data using our private key, send the signed data to the destination, and the signature can be only validated by using our public key.

The key pair mentioned are stored in a wallet. These wallets facilitate the management and even prevent the loss or theft of the key pair. It is important to notice that if the public key is lost, the user cannot identify which transaction he owns. Moreover, if the private key is lost, the user loses the ability to sign, thereby he cannot make more transactions.

As mentioned, transactions represent a transfer of some sort of asset between two key pair owners. The data within these transactions highly depends on the context of the application. The transactions must always be signed, and other nodes verify their signature. If valid, eventually it is spread across the network and put into a block. Otherwise, it is discarded. Then a consensus algorithm accomplishes the process where it decides what block will be inserted into the blockchain. Next section will be addressing several types of consensus mechanism currently used.

So far, we mentioned the authentication and validation of the blockchain network. It exists two more essential properties: transparency and immutability. Transparency because every block can be searchable, and it presents all the contained information about every transaction made. Immutability because once a block is made, it stays into the chain for all its lifetime. Although, in some practical cases, this immutability does not work entirely, as mentioned. For instance, the Bitcoin's blockchain waits for six blocks to be generated after a transaction, for that transaction to be immutable. However, immutability is one of the core principles of the blockchain.

3.3. Consensus Algorithms

The core of blockchain technology is the consensus algorithm among nodes. Bellow, we present some of the consensus algorithms more frequently used:

- **Proof of Work (PoW):** It uses workload as a safeguard. It means that the network proposed a problem and rewards the node that spent the necessary resources to solve it. By spending the resources, which has an energy cost, it assumes that it is a legit participant because it paid the price. The concept was first presented in 1993 by Cynthia Dwork and Moni Naor on paper "Pricing via Processing or Combatting Junk Mail"[43], later formalized in 1999 by Markus Jakobsson and Ari Juels on "Proofs of Work and Bread Pudding Protocols"[44].
- **Proof of Stake (PoS)**[45]**:** It is the most common alternative to PoW. It tries to solve the PoW problem of wasted resources. It is asked to the participant to prove their own certain

amount of currency to generate the next block. The higher is the stake, the higher is the probability to generate the next block. Presumably, this prevents the users from creating forks because they will devalue their stake. Some authors[46] defend that PoS is not an ideal option for distributed consensus because it raises the problem of "nothing-at-stake". Basically, it means that the block generators have nothing to lose by voting for multiple blockchain histories, thus preventing consensus.

- **Proof of Authority (PoA)**[47]**:** PoA is a consensus mechanism that centralizes the decision to generate the next block. Basically, one node has a particular private key that generates the next block and all other trusts. This type of consensus is the most suitable for private networks. However, if the authority to generate the next block is given such to one node, if it is compromised, so it is the blockchain raising the problem of a single point of failure.

Algorithms such Proof of Brun (PoB) and Proof of Space also exist and implement into blockchains associated with cryptocurrencies. However, it contains limitation and concepts not fully developed for mass use. Several more algorithms exist, and more are being developed However, it not the goal of this numbering and review all.

3.4. Top-Level Mechanism

Naturally, as every technology builds until this moment, its development is made on continuous grow of layers stacked on top from each other. The next two subsections presents are an abstraction layer built on top of the blockchain. The smart contract which is a more sophisticated type of transaction, and the oracle, which is a service that provides input data for the blockchain network.

3.4.1. Smart Contracts

As Nick Szabo, the creator of this concept defines as "a computerized transaction protocol that executes the terms of a contract"[48]. We introduced transaction on Section 3.2 and its characteristics how they are authenticated, validated, immutable and transparent once they are inserted into the blockchain. Now we are introducing a top interaction layer, that executes several different transactions corresponding to specific events.

The ledger used by smart contacts triggers transactions automatically when the programmed pre-defined conditions are met. These predefined conditions are basically contracts term, that the parties involved agree on, translated to code, and associated with an address. It is stored and self-enforced by all the nodes within the network. Sending a transaction to that address containing specific value will produce a deterministic behaviour that will always be predictable. Figure 6 is an adaptation from the report[40] and shows the decoupling layers from the application layer, where smart contracts are inserted on to the underlying connectivity of the internet layer.

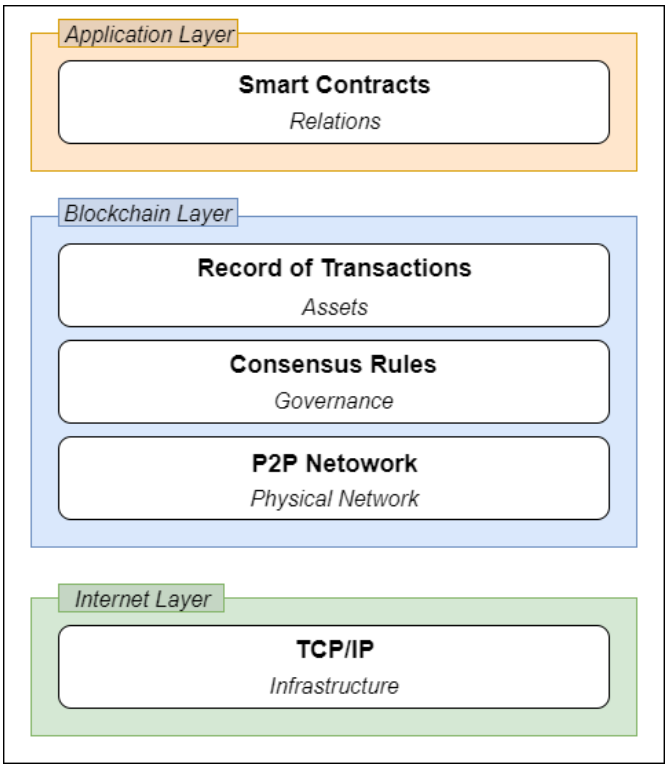

Figure 6 – Stack layer from smart contract application to internet connectivity.

From bottom to top, we can observe that the base of every connectivity on the Internet layer is the protocol TCP/IP presented as green. The blockchain layer is represented in blue, with all the modules we mentioned in Section 3.2. The top located on the application layer, we have the smart contracts representing pre-defined terms to be executed.

As this article[49] stated "contracts are embedded in digital code and stored in transparent, shared databases, where they are protected from deletion, tampering, and revision. In this world, every agreement, every process, task, and payment would have a digital record and signature that could be identified, validated, stored, and shared." It can reduce a lot of cost and time-consuming

process by eliminating intermediaries like lawyers, brokers, banker and probably some public administrators, bringing advantages to a wide range of entities and persons.

Although a smart contract is an agreement between parties, exist potential to be used as legal documents accepted by governments, courts or law enforcement. However, currently should not be confused as such.

3.4.2. Oracles

Blockchains networks cannot access data on his own outside the context it operates. Thus, an Oracle is a third-party service that provides a data feed to smart contracts. As stated in [40] "is an agent that finds and verifies real-world occurrences and submits this information to a blockchain to be used by smart contracts."

As explained in the previous section, smart contracts trigger when pre-defining condition reaches a specific value, which changes its state and atomically triggers an event on the blockchain. Thus, the main task of an Oracle is to provide these values to the smart contract in a secure and trusted environment. Figure 7 shows the Oracle as middleware between the blockchain network containing smart contract and sources of information.

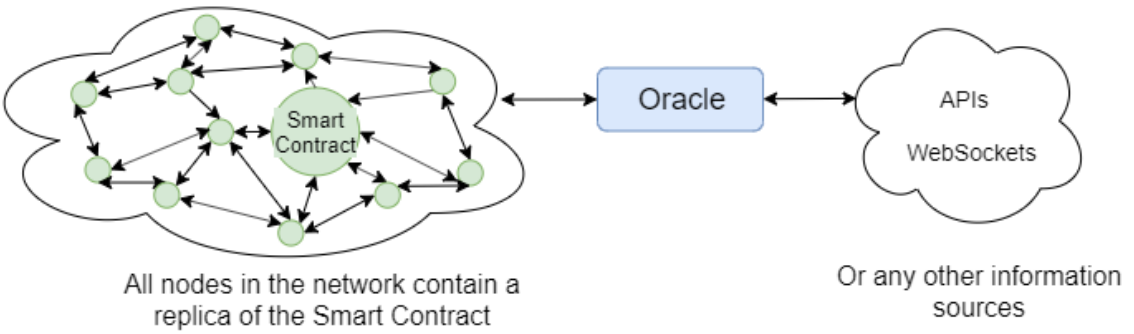

Figure 7 – Oracle communications environment.

This concept is not fully developed and comes with some pitfalls. Since Oracles are third party services that are not part of the blockchain's consensus mechanisms, they are not subject to the security and properties of blockchain discussed before. For instance, the communication between the Oracle and smart contract may be compromised. However, in case we can guaranty that such an

attack cannot occur, the Oracle must fetch information for some data provider. Which although the Oracle may assume the provider is legit and may cross information from several sources, there is no way the validate the information on a deterministic way on a non-deterministic world.

3.5. Types of Blockchain

As mentioned, the first application implemented with blockchain technology was Bitcoin. Since that time, the protocol is open source, and anyone can copy it and start their own version of a P2P network with a consensus protocol. Soon a community start to envolve around this concept and start to conceptualize new ideas, mainly for application with some agreement such as P2P energy trading, P2P carsharing and similars.

Naturally, private institutions also realized they could take some core properties and create their own version. One property, in particular, is the blockchain is permissionless network, which means that everyone can join and starts verifying. For some of these private institutions, this is a security concern, thereby a new type of blockchain arise, the permissioned blockchain. In this concept, exist a network owner which decide the members that are allowed to join the network to verify transactions.

However, some authors defend the "permissioned blockchain" as a controversial term. Thus, the term Distributed Ledger Technology (DTL) emerged. A term that aggregates concepts around distributed ledger, which include the permissioned network and permissionless network also knows as the blockchain.

3.5.1. Public Blockchains

As the name suggests, anyone can join the network without permission and has read and write access.[40] It is a purely distributed network and does not require any central or trusted authority. Also, anyone can send a transaction to anyone, and anywhere in the world, the limitations are virtually and not physical.

Moreover, if the transaction is valid, it is expecting to see it included eventually into the blockchain. Every transaction is transparent, which means that anyone can read transactions on a public block explorer service. Privacy can be a concern. However, anyone cannot be associated to

an address directly, thus being pseudonymous. Furthermore, compared to today standards, the transfer of the assets is faster.

3.5.2. Private Blockchains

At the core, the person or entity implementing and eventually maintaining a private blockchain make every decision that may not benefit all his participants in the same way. “Write permissions are kept centralized to one organization. Read permissions may be public or restricted”[40] Also, to verify the transaction may be appointee to a restricted group that may follow priorities or set of rules, which led to a faster verification than the public blockchain.

Some may argue that a system with these characteristics are not blockchains. However, defining what is and what is not blockchain, currently is hard because it is still in the early stages. As well many argue that dispute between private and public blockchain will eventually follow the same path that networks follow in the 1990s, where the companies prefer to build their network, LAN’s or WAN’s, instead to use the public network, the internet.

3.5.3. Consortium Blockchains

This type of blockchain comes to fulfil a gap between the two polar solutions presented, public and private blockchain. Naturally, the more decentralization exists, more time and resource are needed to reach a consensus. Also, not everyone needs to know about every transaction made into the network.[40] Instead, the transaction verification and validated is perform by a pre-selected set of nodes and transactions itself may have a certain degree of confidentiality.

Naturally, performance and decentralization grow in inverse ways. This type of blockchain allows faster verification than the public blockchain but slower than the private assuming that have other similar characteristics, such as the same consensus algorithm.

3.6. Blockchain Frameworks

Searching the available literature for a blockchain framework, easily we find papers about open-source platform associated with some kind of cryptocurrency token and promoting it to build applications. Although, at the current time, many still in beta testing or with limited functionalities. The most common appearance is Ethereum[39], XRP[50], EOS[51] and Stellar[52]. However, we found also other frameworks without promoting token usages such as Corda[53] and Hyperledger[54], which is a collaborative community that host several frameworks and tools to advance cross-industry blockchain technologies.

Following sections, we will preview two frameworks. Ethereum, from all literature about blockchain and smart contract, it is the most mentioned. Partly because it was the first platform present such concept and for having the largest community in the space. Moreover, Hyperledger Fabric, that does not appear to be backed by any token, instead is hosted under the Linux Foundation Projects[55] which is a reliable organization. As this foundation claim to support a sustainable open source ecosystem providing financial, intellectual and infrastructures resources. “The future of critical technologies such as Linux, Node.js, Hyperledger, and others can’t be left to chance. They need a neutral, independent organization to manage the infrastructure and sustain their communities over the long term.”[56]

3.6.1. Ethereum

Presented in 2013 by the white paper “A next Generation Smart Contract & Decentralized Application Platform”[57] by Vitalik Buterin, Ethereum has been active and contributing to the development and implementation of decentralizing application. As stated on the white paper “The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important.”

The environment is rich in tools and tutorials to build several applications are abundant as we can observe on “get started” official page[58]. The core innovation as became known as Ethereum Virtual Machine, a Turing-complete software that runs on the Ethereum network. Meaning that can

recognise and decide according to the data value and rules sets. Presenting versatility on the use of programming language for potential create a decentralized application on a single platform. However, also making the network susceptible to vulnerabilities that can be exploited, which raise security concerns.

The Ethereum allows the users to store the contracts in the blockchain ledger, and by executing transactions, it will change the ledger state into a determinism value. These smart contracts, also express the proprieties: authentication, validation, transparency and immutability. Allowing the network participant to verify how they will execute and keeping a record that permits to verify independently is the system has been functioning correctly or being fair.

3.6.2. Hyperledger

Hyperledger "is an open-source collaborative effort created to advance cross-industry blockchain technologies."[59] It contains several frameworks for blockchain development as well as tools to help the development and testing. As we can observe in [59], many do not start yet, others in progress and few on the active state. Among them, we find Hyperledger Fabric[60] and Hyperledger Composer[61], which seems to have a vested interest to our goals. Below we present a description for both frameworks.

Hyperledger Fabric

The goal of the framework Hyperledger Fabric is targeting any business that can benefit from a distributed environment. It presented an extensible blockchain platform with several pluggable components to support the complexity that exists in business relationships. As it is claimed, "Hyperledger Fabric is a platform for building distributed ledger solutions, with a modular architecture that delivers high degrees of confidentiality, flexibility, resiliency, and scalability."[59]

For an enterprise use case, the lack of confidentiality is a serious concern with implication on competitive advantage. Hyperledger Fabric tries to find a balance between confidentiality and transparency. It is possible to increase or decrease the level of access of a network participant according to his role and also according to the transaction been executed. It implements a module where it is defined access control base on rules. In other words, this solution allows the implementation of a private or public solution with the same difficulty. It only needs to change the rule set in place without compromise confidentiality, integrity, security and authentication.

In contract, to several blockchain platforms, Hyperledger Fabric does not rely on any native cryptocurrency, either require code to be written on specific language. Figure 8 shows an overall architecture adapted from [62].

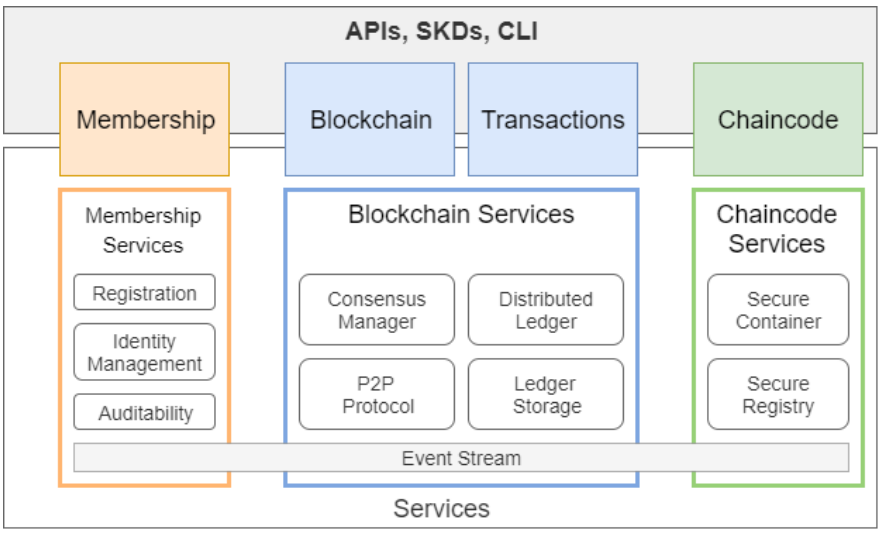

Figure 8 – Hyperledger Fabric architecture.

As we can observe, we find three different services: membership, blockchain and chaincode. The Membership service is used for permissioned model, which can be integrated with the industry's standards. The blockchain services integrate all the underlying mechanism discussed in section 3.2. The chaincode is the smart contracts that contain the business rules of the system. To interact with these modules, the standard way of communication is provided through API's, Software Development Kit (SDKs) and Command Line Interface (CLI).

Hyperledger Composer

Hyperledger Composer is part of the set available to implement blockchain-based solutions. It is a top layer application to interact with Hyperledger Fabric that provides simple configurations for straightforward approaches. To accomplish this, Hyperledger Composer supports the Hyperledger Fabric runtime. Figure 9 shows an adaption from [54]

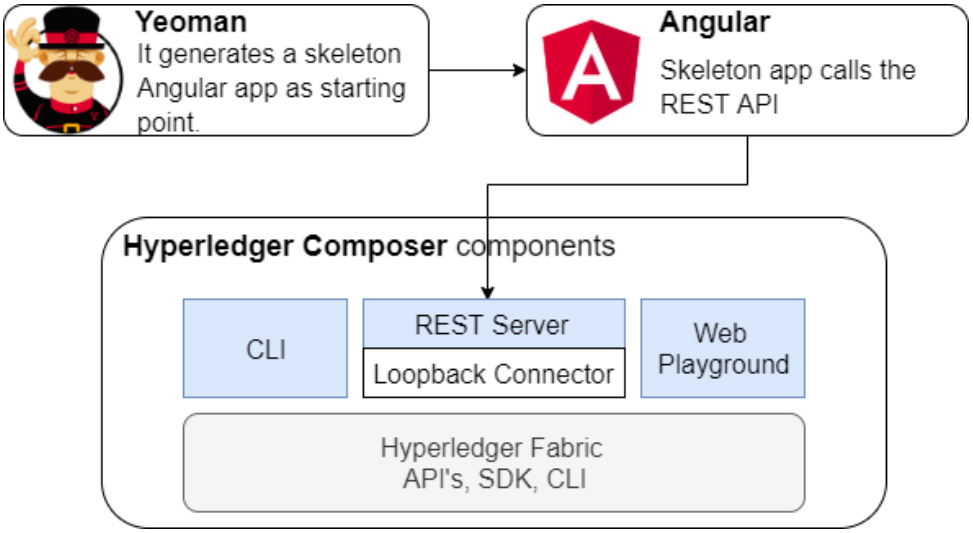

Figure 9 – Hyperledger Composer architecture.

The goal of Hyperledger Composer is to offer abstractions and an environment to more easily test business processes and logic. It can generate a frontend application through project generator Yeoman[63], providing a solution for a time-consuming process. Furthermore, it contains a tool that enables the rapid development of high-level components such as a server providing RESTful API, Web interface and script to containers management. Also, thought the SDK, it is possible to define chaincode, e.i. smart contracts, in an accessible language Javascript[64], and it supports extensions for the popular editors.

4. Solution for Blockchain assisted e-Auctions

By taking advantage of the same business model presented in Section 2.3, that P2P Lending system used, we aim to decentralize business opportunities for a broader range of users. Providing to entrepreneurs with immediate liquidity and give investment diversification to investors. Bring them closer, and ideally presenting better rates or opportunities.

Our main goal is to develop an architecture where will create a representation of asset with the underlying integrity secured by blockchain technology and smart contracts. The assets negotiated between the stakeholder is a receivable asset through auctions. Moreover, all interaction with the system may be accessed and verified through the immutable transaction record.

Algouh, in our context, is present an economy that has been developing a financial ecosystem with regulatory entities and large corporations over centuries, it is out of our scope foresees legal and political procedures for accounting and taxations.

4.1. Scenario of Application

Presented in Section 2.1, the solution used by the traditional banking system to pay management debts between stakeholder, it is thought the creation of financial instruments, typically, made from a large pool of assets. The bonds or loan are some examples presented to manage receivables assets and create liquidity. However, this solution can mix a good obligation debt with a bad one, despite the procedures involved to prevent it, and not always are available to everyone.

Our context of application thought this chapter, will focus on assets that are some other kind of receivable asset. However, the target stakeholder follows the same model of P2P Lender platform. Within the scope of this work, we will not create any new financial instruments, and we will treat each stakeholder and asset as unique.

The following sections will introduce the entities involved in our solution, an introduction of the current centralized scenario and lastly an introduction of a top-level view of our solution.

4.1.1. Entities

Among stakeholders, are include Small and Medium Enterprise (SME), who are looking for a liquid asset e.i. cash, and willing to trade for receivable assets they own, e.i. debt. Thus they present an investment opportunity by selling an asset through auction. We assume that the assets they hold are from services or products they provide to other SME. However, this second SME, which essentially was the buyer of such services or products, will not interact with our solution.

The investors are anyone willing to provide a liquid asset for later repayment. They analyse and accept the risk involved and bid among each other into our platform to acquirer the investment opportunity.

4.1.2. Centralized scenario

The SME (seller) providing services or products to others SME's (buyers), the seller acquirer a receivable asset which is collected over time. Typically, the payments may arrive between a period of 30 to 90 days, as well may arrive in the form of instalments payments over a period. This process is a natural relation which exists in the business world, and both cases may contain interest or not.

Within this typical and straightforward interaction, we identify a difficulty of the SME providing service or products to receive liquidity, when interacting with other SME or even public institutions — creating a short or medium-term debt.

Some solution exists, provided by financial institutions. These institutions can offer to provide immediate liquidity because of their natural access to large amounts of money. In some cases, this process is integrated within the partnership that SME's have when associating they business with financial institute, e.g. banks. For the SME buyer, this process is transparent.

However, in the context of SME, not every obligation debt is legible to be financed in advance by these financial institutions This situation may occur because of the small amount of value these companies operate on, and the financial institution needs more significant margin to have profit. Or even the interest rate may be high enough for that SME seller decide that will not pay off go through the logistics and time spend to receive the desired value.

Despite some small amounts of cash they operate on, SME's exist in greater quantity in the global economy, which means adding an automatic and effective process through an information

system, it can operate in a high-volume transaction and low-profit margin and may be capable of generating self-sufficient economics.

4.1.3. Distributed Scenario

By reusing the example explained above in Section 4.1.2, the short or medium-term debt can be liquidated using other sources. We can apply the same the P2P Lending platform features so far mentioned: a broader range of users, automation process, cutting in logistics; accessible information in real-time; taking advantage of the information system.

The core goals and objectives are the same as a centralized solution running by financial institutions: presenting liquidity. However, with different stakeholders providing liquidity and probably some advantages such as the better rates Figure 10 provides a simple top view of the operation in context scenario.

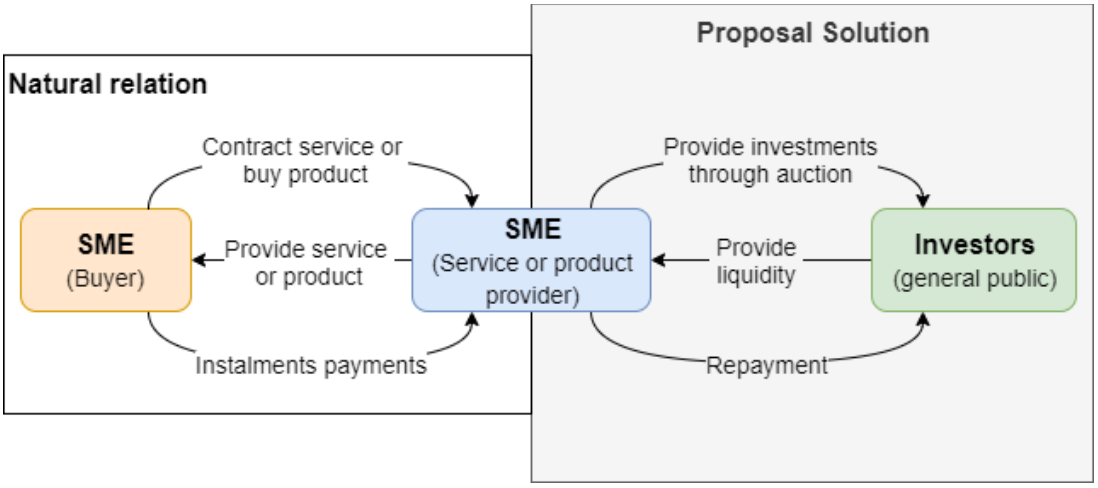

Figure 10 – System's top view operations.

We can observe two environments, at the left, the natural relationship that occurs in the business world to conduct exchanges of goods — two enterprise business, on orange and blue, representing a buyer and a seller respectably. On the right side, we presented our proposal interaction between enterprise business and investors that will occur in our solution.

4.2. Interactions Overview

Our solution tries to find a balance between the relationships among the roles involved: SME's and investors. The resulting information system will act as an observer and provide support to the operations illustrated in Figure 11. The following details the properties of the intended features to each role. Note that one user is not necessarily linked just to one role. The same user may present investment opportunities from a business he owns as SME, and also participate in auctions as an investor.

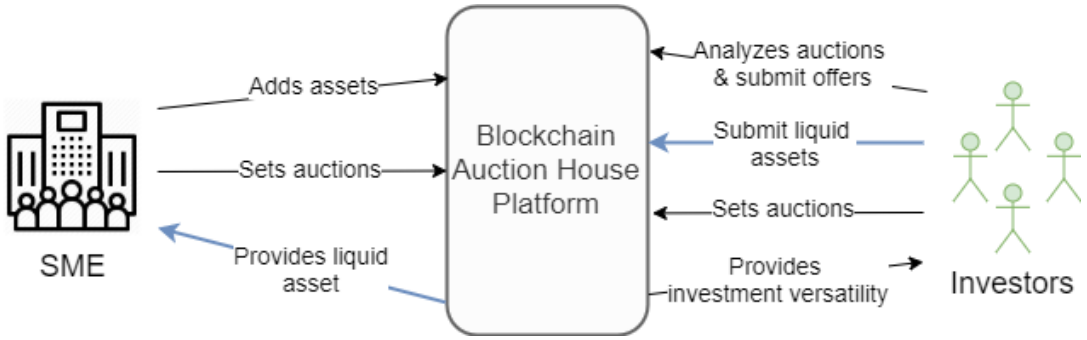

Figure 11 – Stakeholder interactions with the auction management system.

As mentioned, the intended selling method is through auctions. This method ensures that the SME sellers and the investors exchanges assets at the best price possible.

When an investor acquires an asset thought auction, he can submit other auctions as soon he has ownership over the asset, into the secondary market. This option enables the investor to liquidate his assets if needed for whatever reason. However, the asset condition he purchases cannot be modified, except the value, which naturally it will be changed by the auction process. Note that his role does not change into the network.

The disclosure of the user identity (SME or investor) and the asset involved in an auction is always disclosure to promote transparency and better financial decisions. Also, when an investor wins an asset, his identity is disclosure to the auction creator. Furthermore, the auction creator is not legible to submit offers into his auction.

For practical purposes, the resultant solution will not deal with a real liquid asset will simulate the values exchanged; this is represented as blue arrows in Figure 11. Also, the solution will assume the legitimacy and legality of the entities and funds register are accordingly with legal and polities

aspects. The SME seller and SME buyer, naturally know each other for the creation of the receivable asset. However, the disclosure of the SME buyer highly depends on the condition underlying in the receivable asset. Furthermore, it depends on if the contract is with an individual or company. For instance, for individual details, the regulation in place about data protection must be considered, e.i. General Data Protection Regulation (GDPR)[65] in the European Union. Thereby we assume that everything is according to legal terms.

At his core, it will be an engineering approach ensuring, integrity and immutability by the used of an emergent technology blockchain and smart contracts.

4.3. Requirements

The following paragraphs contain the functional requirement to develop a software solution.

Register stakeholder

The platform will provide a registration form to add SME and investors into the system. The registration form will contain the necessary information to be inserted. Such as the name, email, tax identification number (e.g. Corporate Identity Number (NIFC) or Individual Identity number (NIF) for Portugal), credential information, address and activity number in case of SME's.

For practical reason, when registration is performed, a balance attribute must be set which contains the available amount of money to spend in the resultant solution.

Register assets to be sold

Only SME's can insert assets to be sold through the auction into the system. Thus, digital documentation must be provided to prove the ownership of the asset inserted. The blockchain network will not contain these files. It will just have a unique identification that can be traceable to another persistence source.

Define auction

A user may set an auction for an asset own by him and is available into the system. The auction requires necessary information to be scheduled, which are The the auction type, and the dates in

which the auction will open and close. The reserve price, which is the minimum amount the asset has to reach to sold and the bid increment, are optional. An auction will also have a state that will define the permission and operation over it. Details will be presented bellow, on Section 4.5.

Furthermore, to fully take advantage of smart contracts, the user may set the terms to be executed when the auction closes. These terms are translated to code, which is a smart contract may be defined through a template.

Open and close auction

The auction opening and closing processes may trigger automatically soon the set dates and time are reached. Alternatively, the start auction process may be set manually by the user if every information required is set and valid. However, for practical purposes, both process will be available manually.

Furthermore, on the close auction process, the smart contract defined by the user will execute and perform the terms which are translated into code. If an auction does not contain any bid, it will close and the asset will remain untouched.

Accept and validate bids

As mentioned, everyone can submit a bid except the auction owner. Also, not every bid is accepted. For instance, it must respect the bid increment if defined, and the date and time interval in which the auction opens and closes. The bids must be validated by the amount intended to buy the asset, which means for every bid, the bidder must have the balance (set for practical purposes) available for further payment in case of winning. Also, on Section 2.4.2, we mention deviation behaviours and respectively solutions. Our solution also foresees just events may occur, thus further discussed is presented below, on Section 4.3.1 about this matter.

Transfer ownership asset

The goal is the exchange of assets. Logically, it must support a feature to transfer asset between two different users. The auction's asset will be transferred through a transaction, and the balances will be set by normal code execution for practical purposes. This transfer will be supported by the blockchain properties, the immutability and integrity and features, the smart contracts.

Safeguard the history of an asset and stakeholder

As discussed throughout Chapter 3, the immutability and integrity of the transactions records are features of blockchain network by definition. Thus, we can guaranty an immutable record with the underlying integrity secured. Methods to access this information must be available by the administrator of the network as well to the entities that may contribute to the network.

4.3.1. Preventing Deviation Behaviours

As mentioned in Section 2.4.2, the e-auctions have behaviours that violate good conduct of an auction. We also discuss some solutions implemented by the e-auctions platforms house. And our architecture must also foresee and prevent such violations from occurring.

The snipping protection is fairly easy to observe, but it is a complication to define the snipping parameters. The solution is straightforward, if several bids are being made at the last second into the auctions, the close time it will be extended. This parameter has to be defined depending on the delay that occurs on the consensus algorithm.

By default, a successfully placed bid are all bids done while an auction is active, the difference between the last valid bid prices respect the bid increment, and the investor placing it must have the balance available into his account. Also, by default, any bid cannot be cancelled in order to prevent have shield bidding protection. But mistakes happen, an exception must be made when two situations occur.

- The seller does a substantial change into the auction properties or asset being in an auction. Email notification must be always sent to anyone who successful place a bid when such a situation occurs, and bids may be cancelled.
- Client application such present confirmation form when the investor typed an abnormal bid amount considering the last successful bid and the increment required.

Also by default, it is not possible an auction owner bid on his auction. However, if he bid on another account on his auction, it is fairly more hard to detect this situation. A possible solution may be the use of strong authentication mechanism. Also, when a user is registed, we ask for official data like the NIF, which prevent the user by using two or more account with proper authentication. But nowadays, tools to prevent such detection are widely available and fairly easy to use. Thereby, the

first line of defence should be a proper identification account and authentication process after user registration.

Lastly, the two deviation situations presented by eBay and introduced in the last paragraph of Section 2.4.2, transaction interference and interception, can be prevented by not sharing the investor's contact information. Or at least give the option for a user to do not share his contact information publicly. A user when evaluating an auction may see the username about who is bidding and the amount bided, but if he wants to contact some other user, it may request contact information directly to the user.

4.4. Workflows

Upon user registration and respective identity identifications, the system workflow, the business logic workflow, it starts by an SME participant registers an asset into the platform. Then it starts a process it starts for the asset to be confirmed as shown in Figure 12.

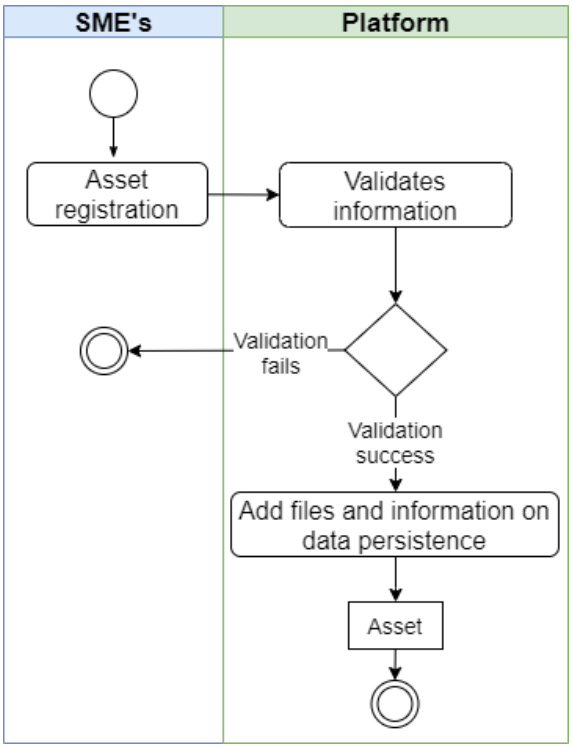

Figure 12 - Asset registration workflow.

The first validation occurs in the registration form present into the client application. However, the verification process represented refers to the mechanism present into the services inside the application business logic. The files added cannot be verifiable by algorithm or at even not in a secure and deterministic way. Although it is part of the engineering process, we must add that the asset should only be valid for auction after a human verification process.

As soon the asset is valid for an auction, the user may create the auction at any time desire. The required information mentioned in Section 4.3 must be set and again, the client application should verify firstly the information inserted. But it does not mean that the platform cannot also perform that verification when the service is called. Figure 13 illustrates the auction workflow inside the solution proposed. As soon it passes through validation process an auction business logic object is created and persisted on the platform.

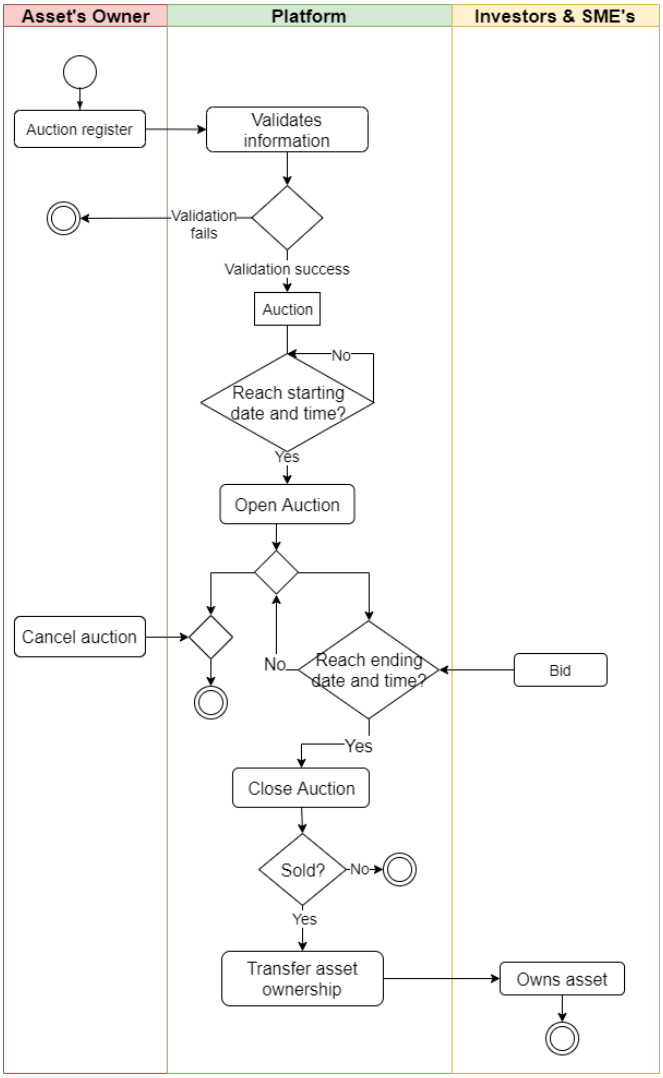

Figure 13 - Auction workflow.

When the starting date and time is reached, an open auction process begins, and at this state, multiple options may occur. The auction may be cancelled at any time for any reason. As mentioned, the platform offers this possibility without judging its fairness. Also, it is when a potential buyer may bid until the date and time to close the auction is not reached. As soon it is reached, according to the bids made the auction will enter into a state Sold or Not Sold discussed on Section 4.5. If sold the transfer of asset ownership takes place and the process ends. The ownership is maintained otherwise.

4.5. Auctions States

The auctions states are basically the auction's life cycle. The change of state occurs automatically according to the transaction made. Figure 14 shows how they can transit one for another.

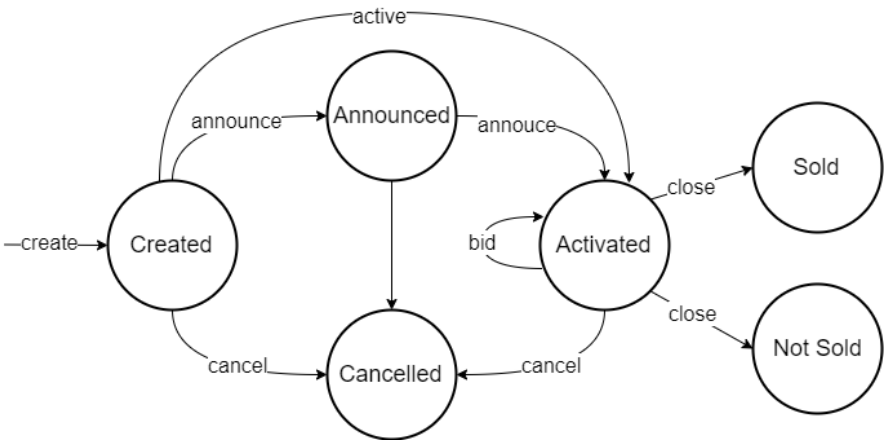

Figure 14 - Auction states life cycles.

The auction, when created for the first time, will appear in the state *Created*, which means that the auctions are only visible to his owner. The required information field needed may not be yet completed, allowing modifications if needed at any time. As soon the auction has valid information, his owner may start the announcement at any time, transiting the state to *Announced*. When the auction is announced state, it will be lock for modifications, and it will be public to all user into the system. At this state, the auction will have information about the date and time open for bids. When that date and time arrives, the auction state will automatically transit to *Activated.* Note that the user may jump the state *Announced* proceeding the auction directly to *Activated* state if he wishes.

All the states indicated so far: *Created*, *Announced* and *Activated*. Allow the user to cancel any auction they own, transit the state to *Cancelled.* The fairness of cancelling an auction after starting without requirements are debatable. However, it is not the goal of this thesis to foresee the implications. This feature is implemented firstly as a possibility that the system allows.

Lastly, when an auction goes through all the processes, excepted the *Cancelled* state, and contains bids, one of two things occurs. The auction does not meet the reserve price if settled by the owner, and he may choose not to sell the asset. Thus, the auction enters into the state *Not Sold*. Otherwise, if the price is reached, and everything ran smoothly the ownership transfer takes place and the auction state is changed to *Sold*.

4.6. Software Architecture

This section, we will conceptualise how our architecture would be connected and describe the function of each module within the system.

4.6.1. Blockchain Selection

To further analyse and discuss the transaction transparency, we need first to know what type of blockchain study on Section 3.5 is needed to reach our goals. To help to define and made a decision we based on the flow chart present in Figure 15 model present by Karl Wüst and Arthur Gervis on a paper called “Do you need a Blockchain?”[66].

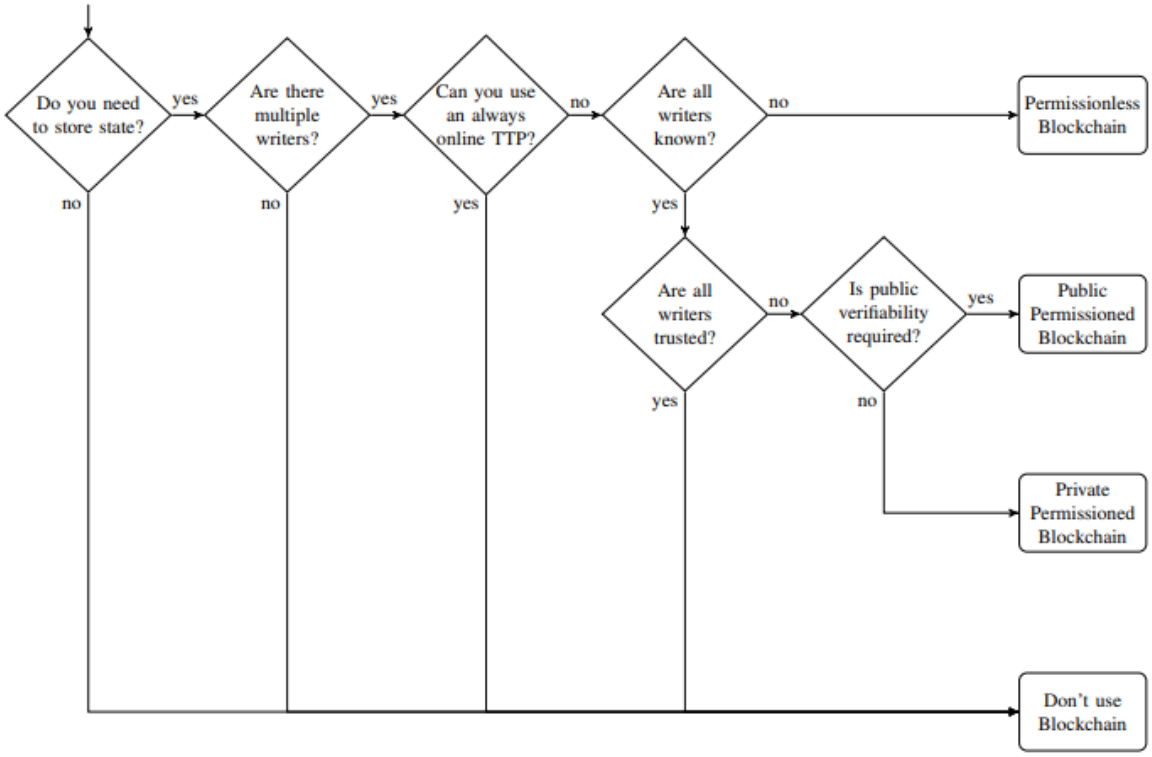

Figure 15 - Flow chart to determine if blockchain implementation is necessary and what type.

Below, we present our answers to the questions present in the chart flow.

- **Do you need to store state?**

  We can easily answer this by mentioned two data structures mentioned that we would constantly manipulate, an asset to be sold and the auctions to sell them. The asset state must always be stored, and the modification that occurs must be traceable to the original state. As well the bids that occur into an auction. We answer yes to this question.

- **Are there multiple writers?**

  As explain through this thesis, and set on initial goals, the intention to open new communication channels to a wide range of investors. Thereby, we will have several users writing into the blockchain network. We answer yes to this question.

- **Can you use an always online Trust Third Party (TTP)?**

  The solution that we are present can be an addon or feature in a larger IT ecosystem. Which in turn, that ecosystem may be trusted or not. Whatever the case, it can be determined for sure and, we want this solution to have independent integrity build at his core. We answer no to this question.

- **Are all writers know?**

  We foresee the participant into the network will be identified, as mentioned exist a registration form before interacting with the system. Although he will not implement any mechanism of cross-reference with government data, on production environment it can by IDP service. Thus, the writers on the network are known and we yes to this question.

- **Are all writers trusted?**

  The writers are primary persons, thus the behaviour although will be restricted, but it cannot be entirety predict. Thereby the writers are not trusted.

- **Is public verifiability required?**

  It depends on the context. In our specific context, it is not correct to let anyone access transactions made. Which imply access to actions state, thus assets and reveal content that is not very private, but it is not so public either. Mean to a specific group of people may have the interest to regulate. For example, if the auctions are following some level of fairness.

At the core of the last question, we must decide if the network is public or private permissioned blockchain. The difference is that if public every entity on the network will access the transaction data and have access to information form other business even without participating in any auction. Otherwise, if private, it is possible to allow an entity to participate in the network, but with access only to a particular auction they issued, or they bid. As mentioned, we decide to develop the solution into a consortium permissioned blockchain network.

4.6.2. Components

Figure 16 presents an overview of where we observe different areas represent by different background colours. The main area, shown in yellow, is our secure environment where all the modules are running. The red area represents a user interacting with the resultant IT ecosystem by a client application. The blue area is the Identity Provider service (IdP or IDP) which provide user authentication. The green area presents the blockchain network, within we observe the Peer, which in turn, contain smart contracts represented as SC into the grey area.

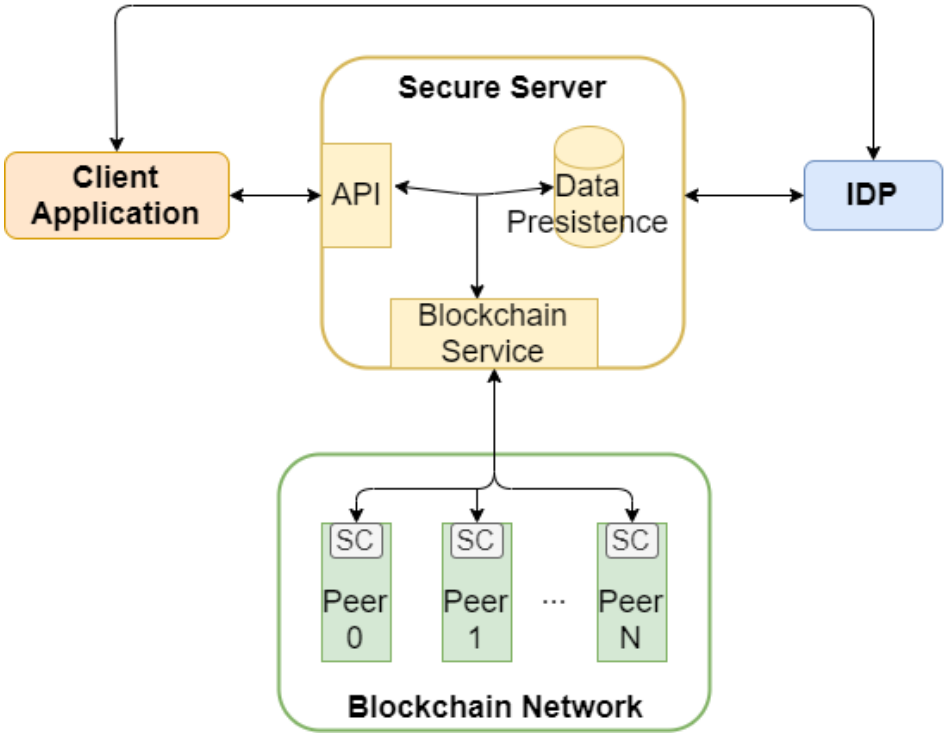

Figure 16 - Software architecture scheme.

Describing Figure 16 from bottom to top, we observe the blockchain network area where all the peers are connected. Each peer may be owned by an independent organization, which may ensure more integrity into the network. Within these peers, it will exist replicas of smart contract, which will be executed with the data distributed by the IT ecosystem, reaching a result and work on the consensus algorithm. For instance, an enterprise participant may want to run his peer to confirm the auction result. This may be particularly useful for public contracts issue by governments to prevent corruption. Also, the context where insurance companies may want to ensure the integrity or authenticity of the contract made may exits.

The layer above, secure environment, is responsible for the business logic. Meaning that is between the user and the blockchain network, providing a framework for the client application through API. Every service call is evaluated if it is a service call, it is for public information or if the call needs authentication or even if it already contains authentication. The data persistence system to save user information and files. The interact with blockchain environment through a blockchain service. Also contains authentication methods to register the new user as well for secure and private communication. The use of an Identity Provider service (IdP or IDP), it can provide certificated and regulated identity to ensure more integrity on the information provided when registering a new user. The authentication architecture in mind is thought a stateless protocol. Thereby after registration, this

server will generate access tokens through delegated authentication protocol that ensures the user will only access the data on his context.

It is not wise to use the blockchain network to store data. Instead, the blockchain network is used as a record, to permanently save the order of each transaction was made on the network. Thereby the use of database server will ensure quick access to the information need for the business logic operate.

Lastly, the client application present in the red area represents a web-based application. Note that each module presented is no limited to one machine, it may be instantiated into several servers working together through virtualization servers and load balances mechanism.

5. Implementation

This chapter, we will discuss the technologies implemented from bottom to upper layer as well as describe our implementation decisions. Starting with relation and non-relation data models, where it was used MySQL[67] and MongoDB[68]. Followed by the Hyperledger Composer framework, defining the implemented transactions and permissions needed and the services for the HTTP requests. Lastly, we discuss the authentication delegator protocol OAuth2.0[69], and the client application for user interaction.

The development and deploy of the further work, it was used the Ubuntu[70] operating system, which is based on the Debian[71], both still in active development. For the implementation of the relational database, we use MySQL alongside with MySQL Workbench[72] which communicates with a web server developed in Java with Springboot framework with the editor Netbeans[73]. For the non-relational database, we used the MongoDB, which communicates with a web server developed in Python3.7 using the microservice framework Flask[74]. The blockchain network was developed with the Hyperledger Composer framework and tools with VSCode[75] and Docker[76] to support the Hyperledger Fabric servers. Lastly, for the web application, it was used the Angular[77] framework and the Github authentication service to generated the token authentication.

5.1. Data Persistence

This section contains the data models used throughout business logic. As the title suggests, it refers to the data persistence module represented in Figure 16 in Section 4.6.

This report[78] presents a comparison between MongoDB performance and features alongside with MySQL Document Store. The test was done on inserting, updating, removing and selecting data. By using different sizes, ranging between 0 and 10^6. It concludes “In every single test we found MongoDB to perform better than MySQL DS”. Also, when the test was performed the MySQL DS was recently released. Thus some performance issues not yet resolved were expected.

The tool chosen was MongoDB, the following paragraph will give a brief introduction to the framework and his characteristics and how it is used into our solution.

MongoDB is a not relation database NoSQL which stores data organize by key-value pairs. Originally developed only as a private tool by DoubleClick, latter MongoDB Inc., to solve scalability

issues with the relational databases they encounter when developing enterprise web applications. Today, it is an open-source project which also has enterprise paid versions and support. Also, with his cloud management service MongoDB Atlas and it is a cross-platform application, meaning it can be installed into the different OS.

5.1.1. Relational Database

The relational database model is represented below in Figure 17, and it is divided into three categories by colours. In green, we observe the data model refer to the users, in red, it is the data model for the user's asset and yellow for the auctions.

The network participants previously identified has two different types of participant: SME's and investors. Both share some attributes which are represented in the table *Users*, identified by a unique number, auto-generated by the MySQL server. Linked to this table are the table *SMEs* and *Investor* through a relation of one-to-one (1-1). Each table, as the name suggests, represent the data for the participant's type with the respective name. They are identifiable by two primary keys, the key from the table itself and the key from table *Users* which is represented as the foreign key in the tables *SMEs* and *Investors*. For practical purposes, the table *Users* also has a balance attribute.

The distinguishes between the *SMEs* and the *Investors* tables are that an SME is a collective person, and an investor a singular person. Particular information needed for these roles should be defined respectively into their class model.

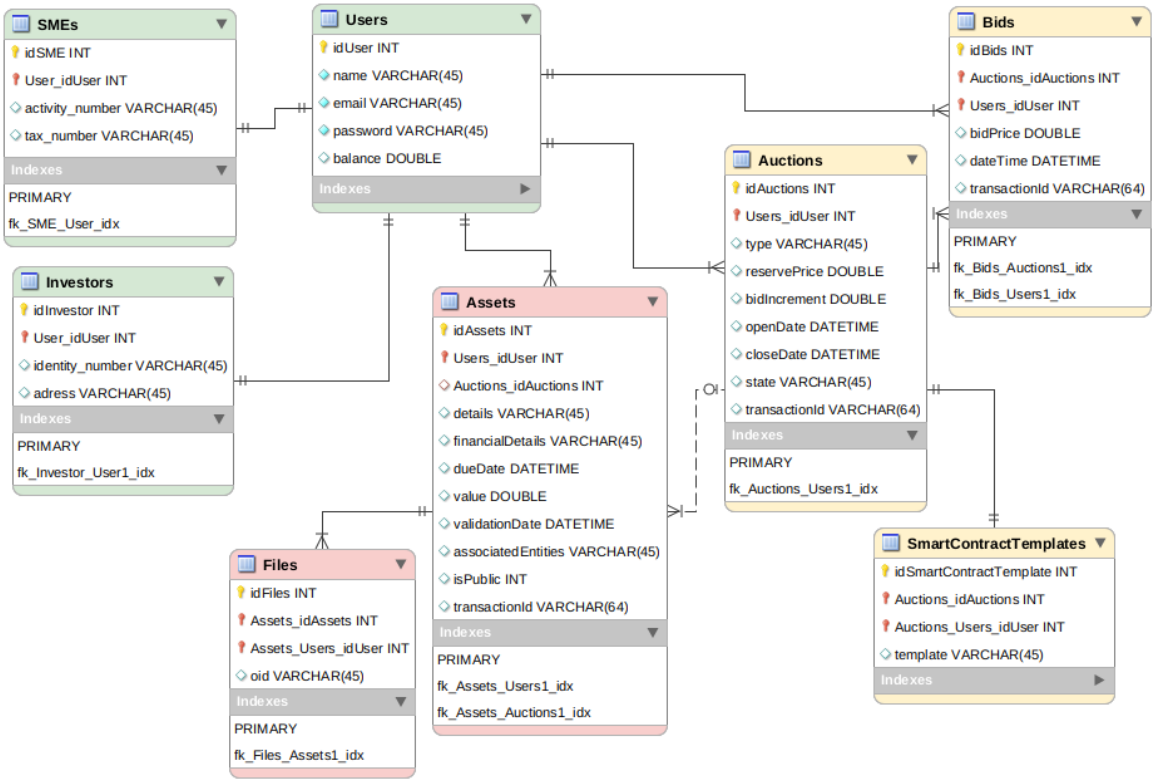

Figure 17 - Demo classes data structures.

The tables *Assets* and *Files* with red colour, represent the objects that the name suggest. Also, as explained before the primary key for the table *Assets* is composed by the auto-generated primary key from the table itself and the primary key from the table *User*. One user may have several assets, thus the relationship between these tables is one-to-many (1-*). The table *Files* contain three primary key, again the auto-generated primary key and the primary keys that provide from the *User* and *Asset* table which are foreign keys in table *Files*. This table just contains the property *"oid"* which is used to store the object identifier of the file into a non-relational database.

Represent in yellow are the tables *Auctions*, *Bids* and *SmartContractTemplates*. Theses tables have primary keys and the foreign key with the same properties that have been mentioned. The *Auction* table also contains the properties needed to set an auction mentioned in Section 4.3 and the property *"state"* which we discussed into Section 4.5. Naturally, auctions have several bids. Thereby, the table *Bids* are linked to table *Auctions* with a relation 1-*. But note that is also linked to the table *Users* with a relation 1-*. The property *"User_idUser"* within the table Bids is provided by the table *User* and do not has the same value that the property also with the name *"User_idUser"* within the table *Auction*. This occurs because the bids are linked to other users rather than the user who created

the auction. Such as we defined on Section 4.3, the user who created the auction cannot bid on his own auction.

The table *SmartContractTemplate* is linked to the table *Auction* with the relation 1-1. The property template contains the smart contract to be exected when the auction close. Note that this property is represented as a varchar type. However, it is just for practical use. On a production environment in a real-world application, this table such be more complex or even had a similar function like the *Files* table, storing a connection identifier to a non-relational database.

Lastly, between table *Asset* and *Auctions*, there is a relation many-to-one (*-1). But this relation is a non-identifying relationship represented in a streak line. This occur because when adding an asset within the system, the asset may not go to auction right away. Thus, the foreign keys *"Auctions_idAuctions"* and *"Auctions_Users_idUser"* will not be set until later user interaction.

5.1.2. Non-Relational Database

Typically, the documents involved in our application may have several tens of megabytes in size, which is not wise to storage into the blockchain. Furthermore, it is not the goal of the blockchain to save the documents. Thus, the need for using some other tool to store files or other types of data is imminent.

Some of the benefits it provides are the high scalability and performance, capable to store a large amount of data for enterprise applications as mentioned. This is possible mainly because a characteristic implemented into this architecture, *sharding*. Which consist of partitioning large datasets into smaller parts for better horizontal escalation. As stated on the possible definition, "based on the idea that as the size of a database and the number of transactions per unit of time made on the database increase linearly, the response time for querying the database increases exponentially."[79] Also, MongoDB does not use tables like relational databases. Instead, they are name collection, which the documents are stored as row but do not has columns. It means that the data structure saved does not have a predefined format. Instead, each row is like a JavaScript Object Notation JSON [80] object, in this way the search performance among row increases because there is no need to join data as the relational databases.

When uploading the files to the MongoDB server, we find the problem, how to avoid malicious alteration to the document? To solve it, we decide to use a one-way cryptographic function based on the SHA-256 algorithm (e.g. hash function), which generates, as a result, an almost-unique 256-bit

or 32 bytes signature for a text. This result is better suitable to compare versions than using encrypted and decrypted methods.

The cryptographic function result also called *digest*, it is anti-tamper by definition. It links the resultant hash to the original text, and at any time or situation, anyone with access to the text can re-hash it and confirm if the text was changed. It is possible because similar texts can produce very different digest values. This method can also be used to data-loss confirmation.

5.2. Blockchain Network

Details about Hyperledger can be found in Section 3.6.2. Throughout this Section, are discussed implementation details using the Hyperledger Composer, and how is possible to be used to start the project and understand its key features to a real-world application based on blockchain technology.

The Hyperledger Composer is a framework build on top of Hyperledger Fabric that hides major complexities and configurations for straightforward development. It provides scripts for simple installation through containers, and management to start, stop and clean the environment when need.

To start, Hyperledger Composer creates a project node based, that needs at least three types of files defined with the model, smart contract and access control. The model file contains the definition of business objects into the network such as the participants, assets and transactions. The smart contracts which it has its modular language, and it uses a smart contract based on JavaScript. Also, the access control file that determines the permissions to all participants interacting with the network.

All these files are putting together on a Business Network Archive file for easy portability, deploy and start network into any machine. Note that while Hyperledger Fabric always needs to implement blockchain solutions, the Composer is an optional component.

As mentioned, the Composer is built on top of Hyperledger Fabric. It uses the Hyperledger Fabric component instantiated into docker containers. They are fourth containers named: fabric-peer, fabric-ca, fabric-orderer, fabric-couchdb. The fabric-peer is the blockchain node that stores all transactions on the joined channels. The fabric-orderer is the service responsible for creating blocks of ordered transactions. The fabric-ca is the Certificate Authority which means is responsible for managing certificates and permissions for each individual. The fabric-couchdb is built with the CouchDB framework which supports the world state database to support querying operation based on JSON objects.

5.2.1. Transactions

The blockchain works as a record of every interaction made in our solution. Within our context, we identify five particular interactions that change the state of the network for each individual. As discussed in Section 3.2, changes on the blockchain state are made by the transaction.

Figure 18 shows the transactions created with the respective attributes. Every transaction has hierarchic properties which are common among them, the transaction identification and the timestamp. The classes names are also with a suffix *Tx*, an abbreviation used to denote transaction. They are defined on the file that contains all the models in the Composer framework.

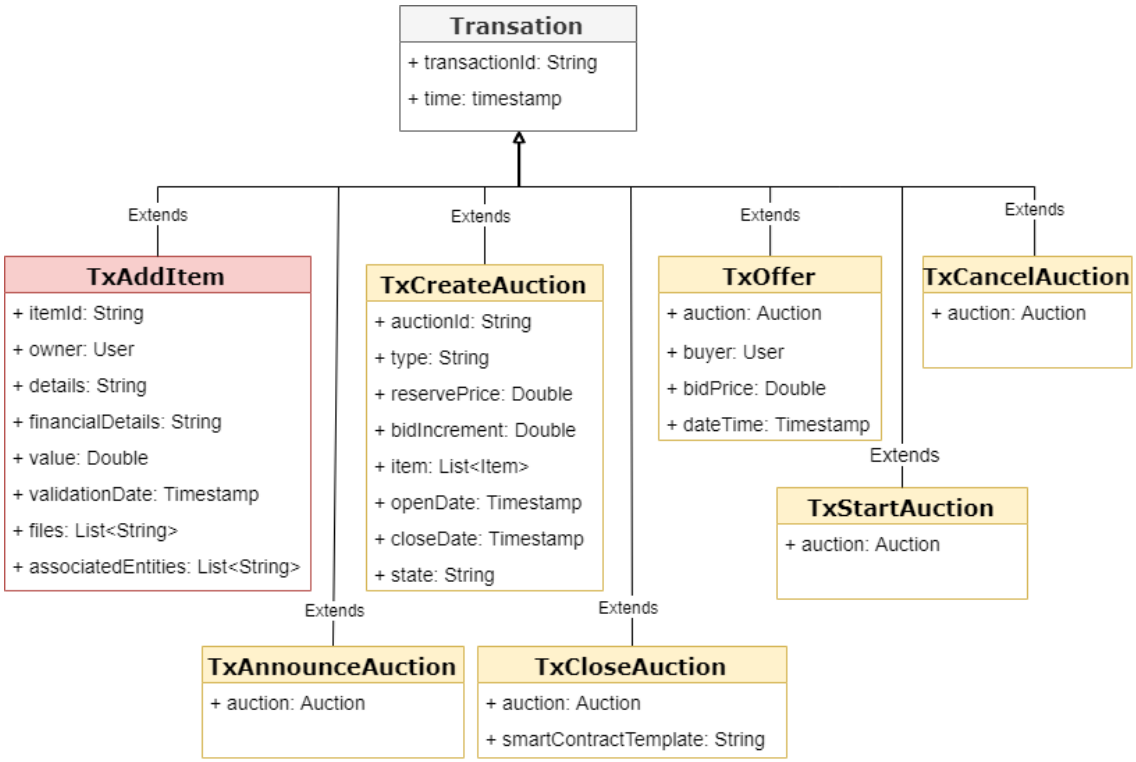

Figure 18 - Demo transaction data structures.

The transaction *TxAddItem* and *TxCreateAuction* will create an asset and an auction respectively into the blockchain network. The *TxAddItem* can only be executed by a participant of the type SME. They are the participant who possesses contracts that can be sold. The transaction *TxCreateAuction* can be executed by both roles, SME and investors. This happens because when an investor acquires an asset, he may need to liquidate the asset for some reason, thus offering also the opportunity to sell.

The *TxOffer* is a transaction that registers the intention of some user in the network acquire the item being sold. This transaction is only valid if the user is submitting it has value enough on his balance to purchase in case of winning the auction. As well the bid is within the range of time that the auction is active.

A participant can only execute *TxStartAuction* transaction successful if the user himself created the auction. Also, the client web application will not present this option to a user that does not own the auction.

For our demo, we give the opportunity to the user close the auction executing the transaction *TxCloseAuction* for a practical purpose. However, in a production environment, this option should not be presented to the user. This transaction is executed automatically by the system when the close time set at the creation of the auction is reached. Or if some malicious deviation behaviour occurs that prolongs the close time, such as the snipping problem.

All auction states are defined by code within the function processor to ensure integrity when they are modified. Figure 19 is based on Figure 14 presented in Section 4.5 with the addition of the transactions that change the auction state.

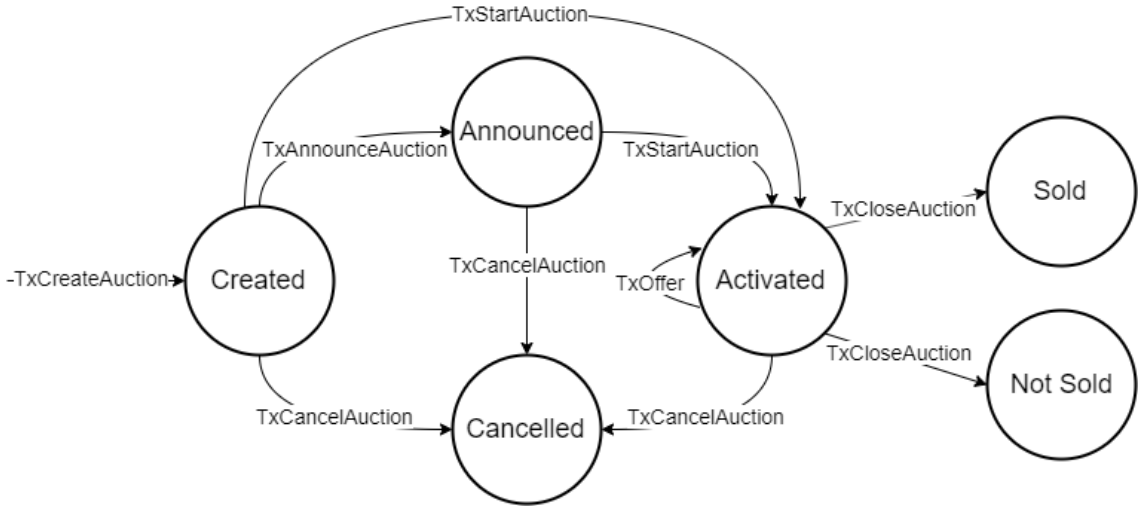

Figure 19 – Auction states changed by transactions

One of the files mentioned at the beginning of Section 5.2 is the file that contains the smart contract based on JavaScript. Figure 20 illustrates an example of a function processor that executes every time that a *TxOffer* occurs in the network.

```
function makeOffer(txOffer) {

    var auction = txOffer.auction;

    // Check auction state
    if(auction.state !== 'ACTIVATED') {
        throw new Error('Listed auction is not for sale yet.');
    }
    // Check buyer balance
    if(txOffer.buyer.balance < txOffer.bidPrice) {
        throw new Error('Insufficient fund for bid.');
    }
    //Check if buyer is the same that asset owner
    if(txOffer.buyer == auction.item.owner) {
        throw new Error('Cannot bid on your own auction.');
    }
    // Check if exist bid increment
    if(auction.bidIncrement > 0) {

        // Sort the bids by bidPrice
        auction.offers.sort(function(a, b) {
            return(b.bidPrice - a.bidPrice);
        });
        highestOffer = auction.offers[0];

        // check if bid increment is valid
        if( (highestOffer+auction.bidIncrement) > txOffer.bidPrice ) {
            throw new Error('Bid do not respect the auction\'s bid increment.');
        }
    }
    return getAssetRegistry(namespace + '.Auction')
        .then(function(auctionItemRegistry) {
            // save the product listing
            auction.offers.push(txOffer);
            return auctionItemRegistry.update(auction);
        })
        .then(function() {
            if(auction.item.owner.company) {
                return getParticipantRegistry(namespace + '.SME')
                    .then(function(smeRegistry) {
                        smeRegistry.update(auction.item.owner)
                    });
            } else {
                return getParticipantRegistry(namespace + '.Investor')
                    .then(function(investorRegistry) {
                        investorRegistry.update(auction.item.owner)
                    });
            }
        });
}
```

Figure 20 - Code example for the definition of a transaction.

We can observe the function *makeOffer* is executed with an input parameter *TxOffer* which is the transaction object define in Figure 18. Before the offer which is a bid, is added to the auction, it passed for conditions to check if it a valid offer.

5.2.2. Access Control Language

The third file mention at the beginning of Section 5.2, it is the Access Control Language (ACL) file that contains the rules that grants or denies the operations: create, read, update and delete into the network. Bellow, it is a list of the rules implemented with a respective short description. Note that some rules within the name stated "Full Access" however, the delete permission is never granted.

- **`Owner Has Full Access To Their Registry:`** The participants are allowed to create, read and update of their registry information.
- **`Owner Has Full Access To Their Assets:`** The participants are allowed to create, read and update of their assets.
- **`Owner Has Full Access To Their Auctions:`** The participants are allowed to create, read and update of their auctions.
- **`Everyone Can Read Auctions:`** The participants are allowed to read auction if the auction is in a state that is different from *"Created"* and *"Cancelled"*.
- **`SME Can Submit Transaction AddItem:`** Allow the participants with the role SME to submit transactions TxAddItem.
- **`Everyone Can Submit Transaction CreateAuction:`** The participants can create the transaction TxCreateAuction.
- **`Everyone Can Submit Transaction StartAuction:`** The participants can create the transaction TxStartAuction.
- **`Everyone Can Submit Transaction CancelAuction:`** The participants can create the transaction TxCancelAuction.
- **`Everyone Can Submit Transaction CloseAuction:`** The participants can create the transaction TxCloseAuction.
- **`Everyone Can Submit Transactions Offer Except Owner:`** The participant can create the transaction TxOffer, except the owner of the auction.
- **`Everyone Has Read Access To Asset On Auction:`** The participants have read access to the asset if it is in an auction.
- **`Everyone Has Read Access To Asset On Auction Doing Offer:`** When the participants are executing the transaction TxOffer, they have read access to the asset in the auction.

- **`Everyone Has Update Access To Auction On Actived Doing Offer:`** When the participants are executing the transaction TxOffer, they have update access to the auction if its state is equal to “Activated”.

Notes that the last two rules are only when a participant is executing a transaction into the network. This case is transaction TxOffer.

Figure 21 presents an example of how a rule is defined. It refers to the last rule mentioned on the list above.

```
rule EveryoneHasUpdateAccessToAuctionOnActivedDoingOffer {
    description: "Allow all participants update access to the auction only when it
    submits offers transactions"
    participant: "org.ua.auction.User"
    operation: UPDATE
    resource(asset): "org.ua.auction.Auction"
    transaction(tx): "org.ua.auction.TxOffer"
    condition: (
      tx.auction.getIdentifier() === asset.getIdentifier() && asset.state === "ACTIVED"
    )
    action: ALLOW
}
```

Figure 21 - Rule implementation example into ACL file.

We can observe that within the rule we define the properties as a key-value pair. We define the participant and resource in which the rule applies, and the operation allowed, in this case, the participant is the User class, the resource is the Auction class and the update operation. Also, it defines the transaction, which is an optional property, if defined means the rule only applies when that transaction occurs. The condition is the logic of the rule, this property can be express in JavaScript code.

5.2.3. API Services

For a client application interact with our solution, it needs some communication framework. The API service is a communication protocol that fulfils that need. By doing HTTP calls, it provides deterministic behaviour and data manipulation by hiding implementation details within the services.

For data manipulation, we need to do essential three operations: create, update, read and delete. Which is translated to HTTP services call with property method set to: POST, PUT, GET and DELETE respectively. The information usually follows define within the HTTP properties. For instance, to send information for the server create a new resource, the information goes into an HTTP call with the properties, method and body set to POST and the data structure, respectively. To update information already into the server, instead of POST is set to PUT. Another common property to used os the parameter, set on the HTTP route. With the API usually always provides also documentation that helps to navigate through the feature provided by the server.

The relational model presented in Section 5.1.1 was used to implemented an API service using Java using NetBeans Editor to develop the necessary code and connection to the MySQL server. The following will show a list of all routes possible to use and his characteristics.

An example of a typical HTTP call is: *https://www.192.168.100.100:6161/{route path}*

For simplicity, the routes will not contain the prefix, IP and port. It presented just the method type and the route path, alongside with a short description.

Investors

- GET: /investor – Fetch data about a list of investors, max of 20 elements.
- GET: /investor/{identification} – Fetch the data about one investor with a matching identification.
- POST: /investor – Creates an investor with data sent within the body.
- PUT: /investor – Updates the investor object with the data sent within the body.

SMEs

- GET: /sme – Fetch data about a list of SMEs, max of 20 elements.
- GET: /sme/{identification} – Fetch the data about one SME with a matching identification.
- POST: /sme – Creates an SME with data sent within the body.

- PUT: /sme – Updates the SME object with the data sent within the body.

Assets

- GET: /asset – Fetch data about a list of assets, max of 20 elements.
- GET: /asset /{identification} – Fetch the data about one asset with a matching identification.
- POST: /asset – Creates an asset with data sent within the body.
- PUT: /asset – Updates the asset object with the data sent within the body.
- GET: /asset /{idUser}/assetlist – Fetch all data about the asset own by an user, matching the user identification, *idUser*.

Files

- GET: /file – Fetch data about a list of files, max of 20 elements.
- GET: /file /{identification} – Fetch the data about one file with a matching identification.
- POST: /file – Creates a file with data sent within the body.
- PUT: /file – Updates the file object with the data sent within the body.

Auctions

- GET: /auction – Fetch data about a list of auctions, max of 20 elements.
- GET: /auction/{identification} – Fetch the data about one auction with a matching identification.
- POST: /auction – Creates an auction with data sent within the body.
- PUT: /auction – Updates the auction object with the data sent within the body.
- POST: /auction/{idAuction}/asset/{idAsset} – Adds the asset with the identification matching *idAsset* to the auction with the identification matching *idAction*.
- GET: /auction/asset/{idAsset} – Fetch the data about one auction that contains the matching asset identification, *idAsset*.

Bids

- GET: /bid – Fetch data about a list of bids, max of 20 elements.
- GET: /bid/{identification} – Fetch the data about one bid with a matching identification.
- POST: /bid – Creates a bid with data sent within the body.
- PUT: /bid – Updates the bid object with the data sent within the body.

SmartContractTemplate

- GET: /smartcontracttemplate – Fetch data about a list of templates, max of 20 elements.

- GET: /smartcontracttemplate/{identification} – Fetch the data about one template with a matching identification.
- POST: /smartcontracttemplate – Creates a template with data sent within the body.
- PUT: /smartcontracttemplate – Updates the templates object with the data sent within the body.

Note that the service for files should only be called when already exists data about an asset into the network that the user making the call previous created. Furthermore, the information *"oids"* is data provide from another service that interacts with the non-relational database. The message dataflow will be discussed in the next chapter.

Non-Relational database services

To develop the services of the non-relational database, we use another framework. The services are based on the micro web framework Flask, written in Python. Following the same methodology for simplicity, the services are:

- POST: /upload – Pushes the data file for the MongoDB service.
- GET, /getfile/{oid} – Fetch the file from the MongoDB with the identification matching the *oid*.

Hyperledger Composer services

The Hyperledger Composer tools, Composer-Rest-Service mention on Section 3.6.2, allow us to generate the services from the network data model defined into the file for that purpose mention in Section 5.2. Also, these when requesting these services were implemented with the authentication token. Details about this implementation are discussed further bellow on Section 5.3.1.

Following the same methodology for simplicity, are listed below the services to execute the transaction in the blockchain network:

- POST: /TxAddItem – Executes the transaction that creates an asset into the blockchain network.
- POST: /TxCreateAuction – Executes the transaction that creates an auction into the blockchain network.

- POST: /TxAnnounceAuction – Executes the transaction that changes the auction state to "Announced" into the blockchain network.
- POST: / TxStartAuction – Executes the transaction that changes the auction state to "Activated" into the blockchain network.
- POST: /TxCancelAuction – Executes the transaction that changes the auction state to "Cancelled" into the blockchain network.
- POST: /TxCloseAuction – Executes the transaction that changes the auction state to "Sold" or "Not Sold" into the blockchain network. It is also responsible for making the transaction between the seller and the buyer of the assets involved.
- POST: /TxOffer – Executes the transaction that shows the intention that a buyer has to buy a specific asset through the auction.

5.3. Client Application

As mentioned in Section 3.6.2, the Hyperledger Composer provides a tool to help generate a client web application based on the Angular framework. Although we tested this feature and generate a skeleton web application through Yoeman application, we do not use it to develop the client web application present on the following Section 0.

Also, the authentication used was provided by GitHub to generate an authentication token base on protocol OAuth2.0 using the passport strategy.

5.3.1. Authentication

All communications are done by the embedded HTTPS protocol which secures the data, ensures confidentiality and anonymity to third parties. Our solution, it also uses a stateless mechanism for authentication for interact with the network, meaning it uses a token stored on the client-side cache (e.g. cookies) to authenticate each service call from the client web application.

For a business organization, the standard enterprise decision is the use of protocol such as Security Assertion Markup Language (SAML) or Lightweight Directory Access Protocol (LDAP). Although the Hyperledger Composer provides a tutorial[81] that uses the OAuth 2.0 protocol, we also consider the JSON Web Token (JWT) authentication. However, OAuth 2.0 defines a protocol

while JWT defines the token format. We find the limitation that JWT does not provide specification on how to obtain the token in the first place and it not backup by a standard. In contrast, the OAuth 2.0 protocol, it uses bearer authentication scheme specified by RFC-6750[69].

We implemented the protocol OAuth 2.0, authenticated by a Google Account, which means that every interaction made by the client can be traceback to that account. Figure 22 presents a flow of communication among all services. Note that instead of Google authorization service another Identity Provider (IDP) can be used, such one that is maintained for governmental institutions. Thus, given proper backup to every interaction with the system.

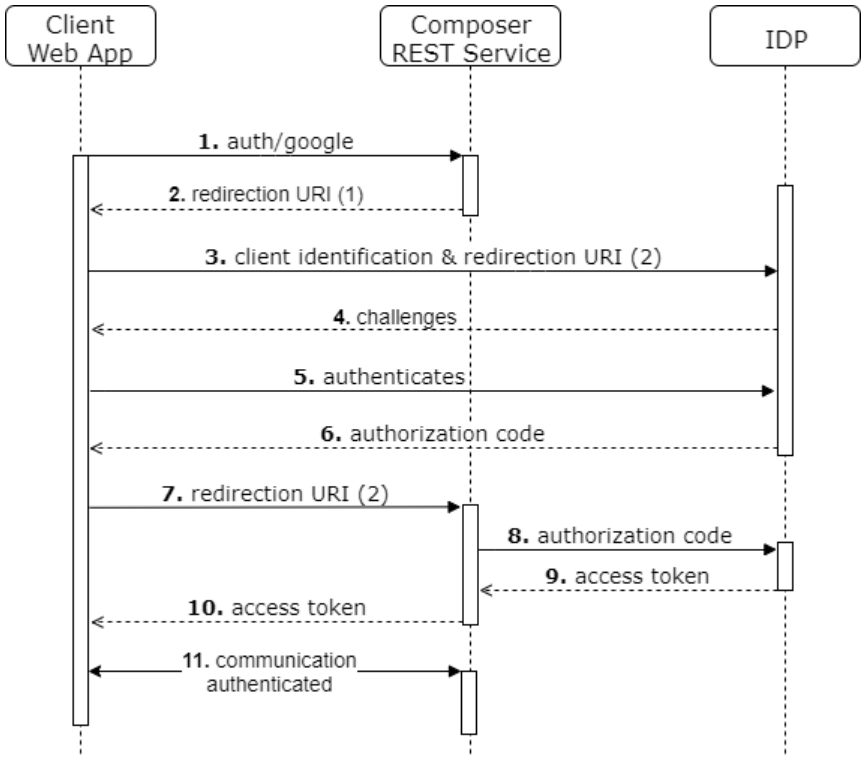

Figure 22 - OAuth 2.0 Protocol flow.

The communication flow starts by calling a specific endpoint which redirects to authentication login page provider by Google API, present by step 1, 2 and 3. Between step 2 and 3, the redirection is automatic, does not exist action from the user. Note that step 4 will carriage two parameters, client identification and redirection URI (2), on the request headers. The first one is the client identification provider when the client web app is registered into Google services. The second is the URL to receive the reply when the challenge is completed.

The login page is presented as response and is inserted the account login credentials, which is the response to the challenge. If everyone succeeds correctly in response is receive an authentication code associated with the account, step 5 and 6. When received the client web app sends a request to composer rest service which redirect to google service to confirm the code. If valid, an access token is received and redirect to the client web app, represented on step 7, 8, 9 and 10. Again the step between 6 and 7 is transparent to the user. If completed with success, all end-points communication to the blockchain network is authenticated, and the user will have access to the information that has permission.

This flow was implemented using Google API services for the sing up an account be authenticated and fetch some information about the user for practical purposes. On a production environment, another certificated IDP service should be used. Note that the step 3, 4, 5 and 6 are indicative, may change if the IDP Provider change.

Additional configuration is needed for the environment variables at the computer system in which the Hyperledger framework is installed. Code 1, is an example of the configuration used to be inserted into the terminal.

```
export COMPOSER_PROVIDERS='{
    "google": {
        "provider": "github",
        "module": "passport-github ",
        "clientID": {Generated Client ID},
        "clientSecret": {Generated Client Secret},
        "authPath": "/auth/github",
        "callbackURL": "/auth/github/callback",
        "successRedirect": "{success url}•",
        "failureRedirect": "{failure url}"
    }}'
```

Code 1 - Environment variable definition for passport strategy.

Note that *{Generated Client ID}* and *{Generated Client Secret}* must be replaced by the values generated when the authentication configuration is completed on Google API services.

5.3.2. Angular

Our demo includes a web application which each end-user can interact through the web browsers. It was developed by the Angular[77] framework and manipulates the system data thought web services and uses OAuth 2.0 presented on last sections.

When building a web-based application, several factors must be considered, mainly usability and user-interface. However, it is not our goal to be a concern with the best possible usability for the end-user. We focus on a practical approach to the interaction of the auction business network. Which do not mean we will ignore simple good practices for an intuitive experience.

The framework chosen was Angular, a TypeScript-based open-source project, mainly maintained by Google. Alongside with Bootstrap[82], a widely known framework to design templates for forms, buttons and other interface components based on HTML and CSS. Figure 23, 24 and 25 are an example of different pages.

Figure 23 shows the home page for two different participants. An SME on the right side, which typically will act as an SME and on the right, the Investors. Note that the seller home page has a slight difference, it has one more button *"Add Item"* and a registered asset *"Asset01"* with one document.

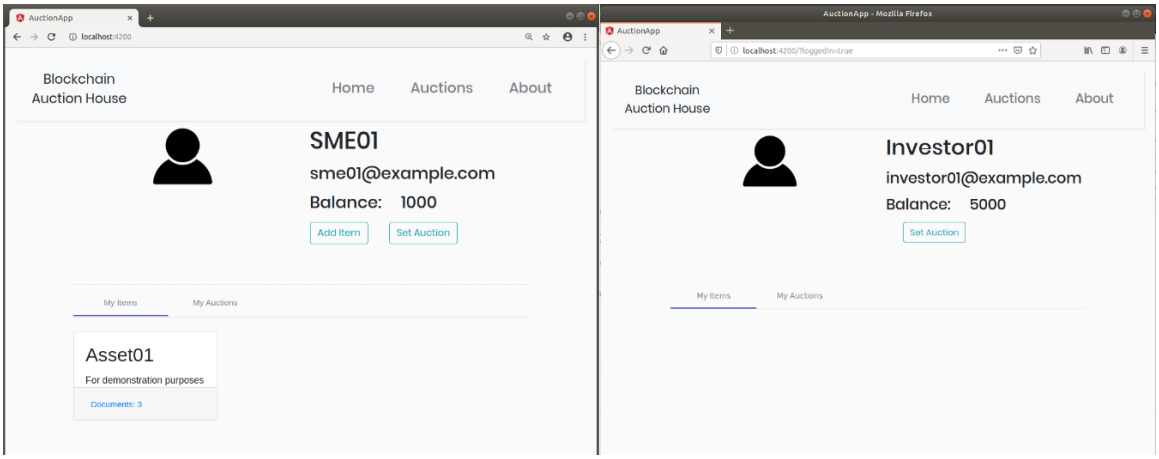

Figure 23 - Demo home page in the right for the seller and in the left for the investor.

Figure 24 shows on the right the seller's home page into the auction tab. Where we can observe the auction on sale, with reserve price define and no offer made. At the left side, it a public

area, accessible by the button “Auctions” in the middle of the top navigation bar, displaying the auction for any participant in the network.

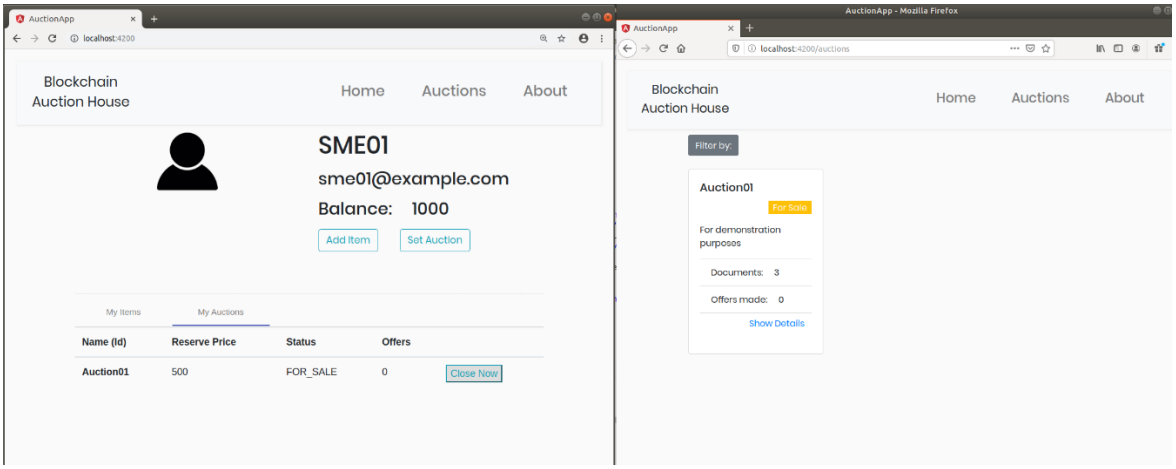

Figure 24 - Demo home page on the auction tab for the seller on the right side and the auction on sale on the left side for the public.

Figure 25 shows the same view for two different investors who bid into an asset. On the right side we have *Investor01* who made an offer bidding 200 and on the left side, the *Investor02* who bid 250. Note that each view does not display the bid from the other investors. Because each investor does not have the permission to know the owner of the bids but was the right to know how many bids were made and what was the amount.

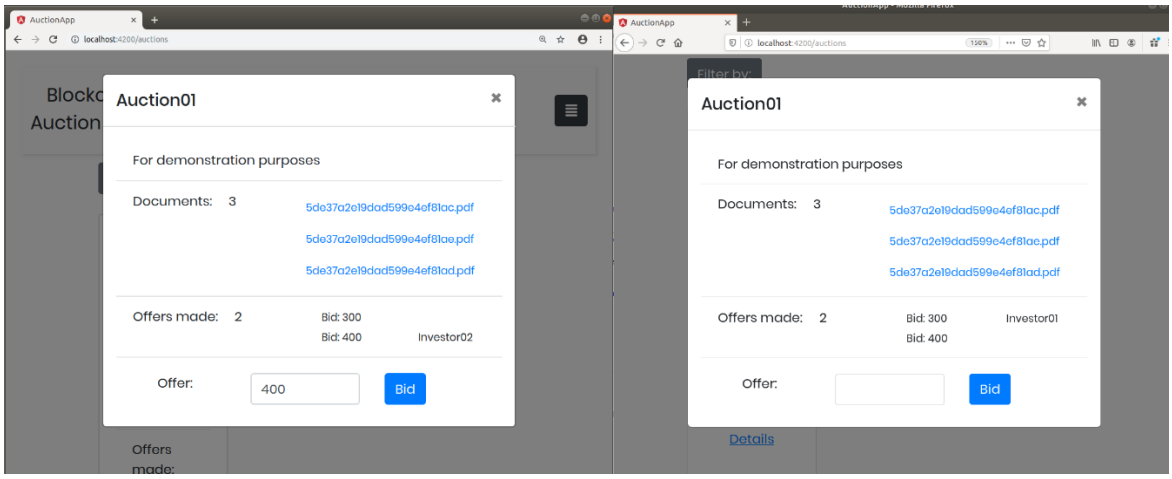

Figure 25 - Demo view for auction details and bid options.

These last views we can also observe, a hyperlink to document publish with the auction that contains details about the asset to be sold. The document name may not be intuitive. However, in a practical sense, the name represents the object identification (OID) generated by the MongoDB when upload by our service developed for theses case. It is possible to download the document by clicking into the name.

The designs presented are not optimized for mobile devices, although it is possible to interact with the demo on such devices when deploying on a production environment.

6. Results and Evaluation

This chapter demonstrates the testing process that our solution went through. Also, present an evaluation and discussion to each test performed., we went through a process of testing it. The results were analysed considering the requirements to conduct an auction defined in section 4.3. Figure 26**Erro! A origem da referência não foi encontrada.** represents the solution environment given the software architecture presented in section 4.6, and the components were instantiated on localhost alongside with docker containers, everything in one machine.

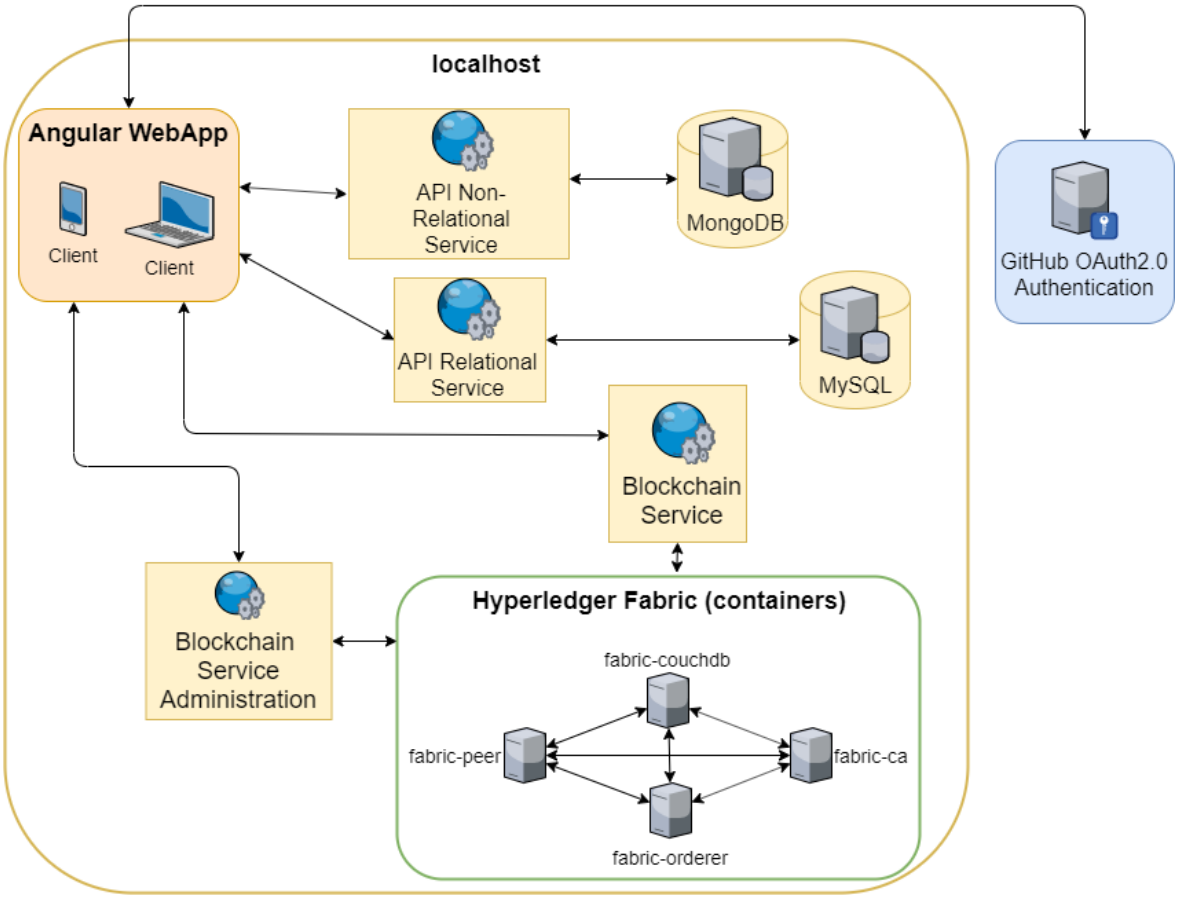

Figure 26 - Solution environment.

Presenting the services starting from the data persistence to the upper layer we have: the MongoDB database server runs on localhost port 27017, connected to the flask base service API Non-Relational Service running on localhost port 5000; the MySQL database server runs on localhost port 3306, connected to the Springboot base service, API Relational Service running on port 2000; the Blockchain Service using multi-user authentication running on port 3000 and the Blockchain Service Administration running on port 3001.

The Hyperledger Fabric containers are also running on localhost. However, each one has a distinct IP address and port. The container *fabric-peer* set with the *IP 172.18.0.5* and port 7053, the *fabric-ca* is set with the *IP 172.18.0.4* on port 7054. The *fabric-couchdb* is set with the *IP 172.18.0.3* on port 5984, and lastly, the *fabric-orderer* is set with the *IP 172.18.0.2* on port 7050.

The web application, which runs on localhost and port 4200, is the composer of all other developed modules. All components are independent, and the web application is the component which provides the service flow, the order for each HTTP request that must be called in order to reach a goal. The following sections will evidence the results obtained mostly by web services request by Swagger[83] or by Wireshark[84], a packets analyser program.

6.1. User Register

To register a participant into the network, independent if it is an SME or investor, the web application must be authenticated. As mentioned, this is done by the delegation authentication protocol OAuth 2.0. When such authentication is not present, an error code appears as a result. Figure 27 shows the error message from an HTTP request without authentication.


```
{
  "error": {
    "statusCode": 401,
    "name": "Error",
    "message": "Authorization Required",
    "code": "AUTHORIZATION_REQUIRED",
    "stack": "Error: Authorization Required\n    at /home/doso/.nvm/versions/node/v8.16.0/lib/node_modules/composer-rest-server/nod
  }
}
```


Figure 27 - HTTP error message without authentication.

As we can observe that the status code is 401 corresponding to “Authentication Required” defined by RFC 7231[85].

When the client web app has authenticated the credentials used by the participant into the network must be generated and associated. This process is done by a sequence of multiple HTTP

requests shown in Figure 28. The same figure shows the order, status, method, destination domain and other details for each request, available from the browser Mozilla Firefox[86] on the network details tab when the operation of register a participant is complete.

Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms
204	OPTIONS	localhost:3001	Investor	xhr	plain	354 B	0 B	2 ms
200	POST	localhost:3001	Investor	xhr	json	590 B	160 B	2288 ms
204	OPTIONS	localhost:3001	issue	xhr	plain	354 B	0 B	1 ms
200	POST	localhost:3001	issue	xhr	octet-stream	1.45 KB	1.01 KB	2498 ms
204	POST	localhost:3000	import	xhr	json	2.37 KB	0 B	13 ms

Figure 28 - HTTP request sequence to generate participant credentials.

We find five HTTP request, the first and third request are *OPTIONS* request methods used to describe the communication options for the target resources which are not crucial for this analysis. In contrast, the second, fourth and fifth HTTP request are essentials to register a participant into the network. The first POST request sends to the server located on port 3001 the information required define on Section 5.2 to create a participant into the network. The participant identification is sent back, and a second POST request is sent, to the same server but a different route, to issue the credentials to be used to sign all the transaction made by the participant.

The REST API is also protected against request with semantic errors. Figure 29 shows an error message when such request occurs.


```
{
  "error": {
    "statusCode": 422,
    "name": "ValidationError",
    "message": "The `Investor` instance is not valid. Details: `name` can't be blank (value: undefined); `email` can't be blank (value: undefined)
    "details": {
      "context": "Investor",
      "codes": {
        "name": [
          "presence"
        ],
        "email": [
          "presence"
        ],
        "balance": [
          "presence"
        ]
      },
      "messages": {
        "name": [
          "can't be blank"
        ],
        "email": [
          "can't be blank"
        ],
        "balance": [
          "can't be blank"
        ]
      }
    },
    "stack": "ValidationError: The `Investor` instance is not valid. Details: `name` can't be blank (value: undefined); `email` can't be blank (va
  }
}
```


Figure 29 - Protection against semantic errors.

As expected, the code *422 Unprocessable Entity* appears, which means the message is well formatted but was unable to be processed due to semantic errors. In this case, the attributes name, email and balance cannot be undefined. The email because is the identification of the user into the network, the name because it is how the participant is identified to other participants and the balance is for debugging purposes only.

6.2. Assets Register

The assets are represented into the blockchain network with documents that prove the existence and ownership. The blockchains are not viable to store the documents due to documents size, inefficiency and mostly is not the purpose for what they were built. Figure 30 is a screenshot from a Wireshark capture, which indicates the packets flow to an SME add an asset into our solution. The asset contains the information defined in Section 4.X with three files.

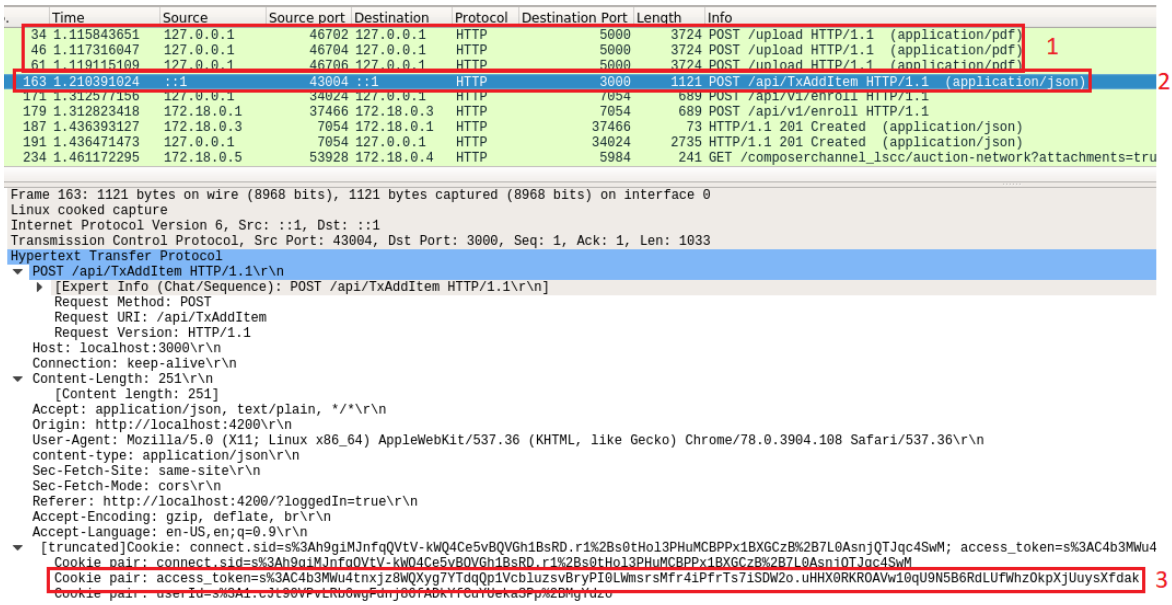

Figure 30 - Wireshark capture, presenting the service requests flow to register an asset.

As we can obverse, we find three red rectangles number: 1, 2 and 3 — each one showing different information. The top rectangle which has the number 1 on the right side shows the first three HTTP request are POST methods, each upload a different file through the non-relational API

service running on port 5000 to MongoDB. Each request sends back an object identifier that is temporarily save on local storage on client-side, to be used on the following methods.

The middle rectangle, which has the number 2 bellow and is selected by a blue background is the HTTP request with POST method which calls the blockchain service running on port 3000. The information sent on this request is illustrated in Figure 18, as *TxAddItem*, which the attribute files are the object identifier save on local storage. The bottom rectangle, which has the number 3 on the left side, contains the information associated with the HTTP request last mentioned. It circumvents the authentication token from OAuth2 Protocol that authenticates the user in the service and allows to sign the transactions only with his credentials.

Soon after the non-relational and blockchain services are completed with success, the final step is to save the information into a relational database. This is present in Figure 31, which is a screenshot from the same Wireshark capture, thus presenting the number 4 and 5.

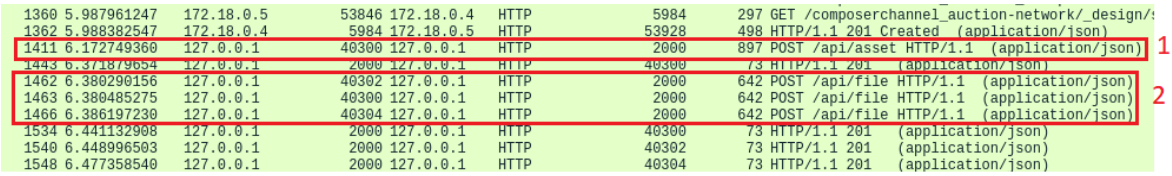

Figure 31 – Continuation of Wireshark capture, presenting the service requests flow to register an asset.

The red rectangles which have the number 4 and 5 on the right side, are the HTTP request with POST method that adds the information defined on the tables Asset and Files, respectively, illustrated on Figure 17 in the Section 5.1.1. As mentioned, it is three files, thus are made three requests to the same route *"/api/file"* but each with different information regarding the object identifier which links to the files.

Figure 32 shows the data class into the MongoDB that saves the documents files.

```
> db.fs.files.find({}).sort({_id:-1}).limit(3)
{ "_id" : ObjectId("5dbaf2173d8176b251b62e45"), "filename" : "asset02_3.pdf", "contentType" : "application/pdf",
 "md5" : "342db690a6ebc1f3c9a72fa592a037e1", "chunkSize" : 261120, "length" : 68891, "uploadDate" : ISODate("201
9-10-31T14:39:19.053Z") }
{ "_id" : ObjectId("5dbaf2173d8176b251b62e44"), "filename" : "asset02_2.pdf", "contentType" : "application/pdf",
 "md5" : "342db690a6ebc1f3c9a72fa592a037e1", "chunkSize" : 261120, "length" : 68891, "uploadDate" : ISODate("201
9-10-31T14:39:19.050Z") }
{ "_id" : ObjectId("5dbaf2173d8176b251b62e42"), "filename" : "asset02_1.pdf", "contentType" : "application/pdf",
 "md5" : "342db690a6ebc1f3c9a72fa592a037e1", "chunkSize" : 261120, "length" : 68891, "uploadDate" : ISODate("201
9-10-31T14:39:19.031Z") }
```

Figure 32 - MongoDB files registries.

For practical purpose, only the request to the blockchain service is authenticated. The request to the services for relational and non-relational persistence do not contain any security processes. Note that only the participant with the r SME role can register assets into the blockchain network., which goes in line with the rule *"Seller Can Submit Transactions Add Item"* present in section 5.4.

6.3. Auctions

All the requests shown throughout this section are authenticated with an OAuth token as presented in the previous section.

6.3.1. Auction Register

To create an auction, the participant must own an asset. This validation is firstly done on the client web application when the form to "Add Auction" is open. Obviously, this raises security concerns because the service can be requested without a web browser, thus injecting data that can break the API server.

Nevertheless, this can be prevented by adding a new rule on the ACL, which contains the rules for the network, by denied service when the transaction *TxAddAuction* is requested with information that does not correspond to any asset that participant has access.

Figure 33 shows the packets exchanges when creating an auction. The red rectangle identifier with the number 1 is the first HTTP request executed by the participant assuming that the web application validates the data when inserted into the web form. The request executes the transaction into Blockchain Service with the information defined in Section 5.2.1. It returns the transaction identification if executed successfully. Otherwise, it will return the HTTP response with code 500, which means a server error, alongside with the error cause.

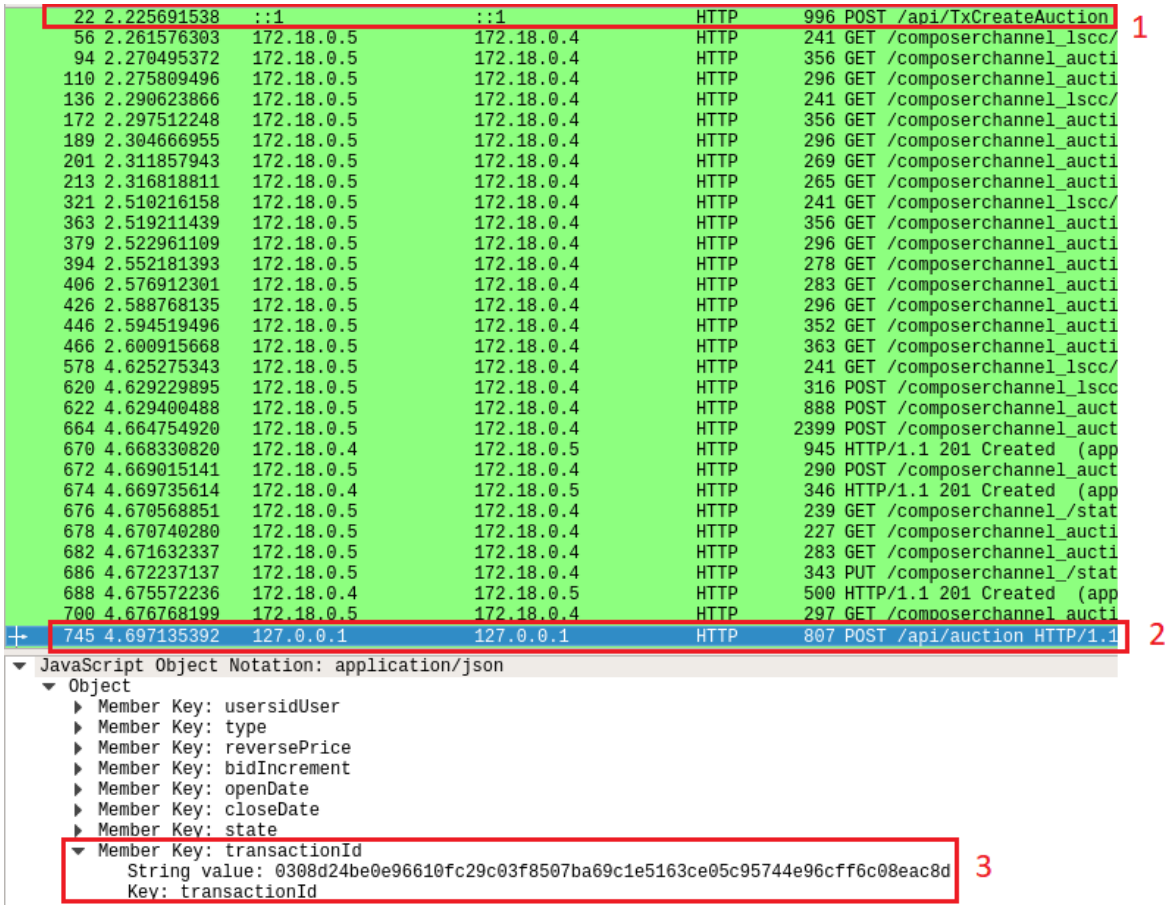

Figure 33 – Packet capture for transaction TxAddAuction.

The red rectangle identifier with the number 2 is the API Relational Service HTTP request to insert information defined in Section 5.2.3. Note that this request is selected with a blue background, thus the information presented above, it is the information carried by the request. The red rectangle identifies by the number 3, surrounds the transaction identifier produced by the transaction *TxAddAuction* successful execution.

Between the red rectangle identified by number 1 and 2 are several HTTP requests. These requests correspond to the blockchain network, more precisely between the containers fabric-peer and fabric-ca. As mention in Section 5.X, the fabric peer is responsible for joining the channel and executing the transactions. The fabric-ca is responsible for managing the participant permissioned.

After the asset creation, the participant can execute the transaction *TxStartAuction*. The methods to autonomous execute this transaction when data and time defined are reach were not implemented. The rules that allow these executions are: *"Owner Has Full Access To Their Assets"; "Owner Has*

Full Access To Their Auctions"; "Everyone Can Submit Transaction CreateAuction"; "Everyone Can Submit Transactions Start Auction"; described on section 5.2.2. Moreover, the data structure of each corresponding transaction is defined in Figure 18in Section 5.2.1 .

6.3.2. Acceptable Bids

When an investor bids in an auction, the services follow a similar flow so far mention. A request is made to API Blockchain Service if successful, follow the request to store the information into the API Relational Service API.

Figure 34 shows a packet capture when the participant does not have the required balance available when it is bidding.

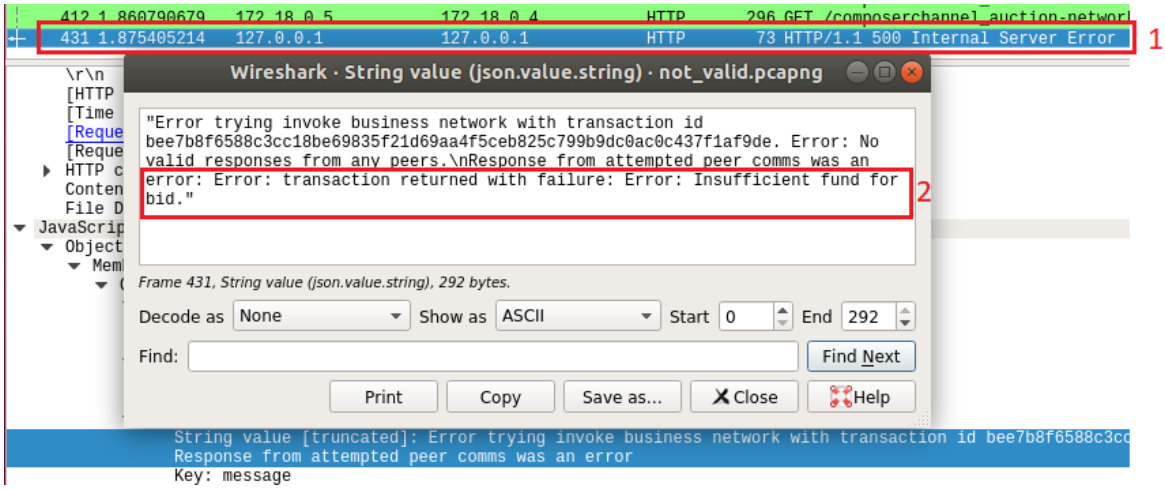

Figure 34 – Packet capture when transaction TxOffer throws an error.

This message can be observed in Figure 20 in Section 5.2.1, alongside with the other validations when executing the transaction *TxOffer*. Such as making a bid into an auction that as a state different than *"Activated"*, or the participant is trying to bid an auction he owns or the bid's value does not correspond to the highest bid plus the bid increment.

Although exist code to check if the bid bellowing to the auction owner, also there a rule preventing such situation defined on Section 5.2.2, named *"Everyone Can Submit Transactions Offer Except Owner"*. Moreover, we can find the rule *"Everyone Has Read Access To Asset On Auction Doing Offer"* because when developing the blockchain network, we notice that although we have

read access to the auction on *"ACTIVATED"* state, we find that we do not have access to add a bid through the transaction *TxOffer*. Thus, when executing the transaction *TxOffer*, it is granted the update permission to the auction to add a new bid. It may sound a security problem to have update access to an asset that the participant does not own. However, the update access is only valid for the transaction *TxOffer* and it is a deterministic result pre-defined by code. The only way to be exploited is somehow modified the code or inject new one.

6.3.3. Asset Ownership Transfer

To transfer the asset ownership, initially foreseen to be performed by an autonomous process that regularly checks if the date and time defined for the auction end are reached. Also checks if the deviation behaviours identified on Section 4.3.1 were being exploited. However, it was an ambitious goal. Thus the transaction *TxCloseAuction* must be performed by the participant that created the auction, in order to determinate the auction winner.

The scenario demonstrated throughout this section is an SME entity, with the identification sme01@example.com, created an auction with the identification Auction01 for the asset with the identification Asset01. Also, the auction did not have minimum increment value or reserve price defined. While the auction was with the state set to Activated, two investors placed a bid. The identification of the investor is investor01@example.com and investor02@example.com. The balance to each participant is zero balance to the sme01@example.com, five hundred to the investor01@example.com and two hundred for the Investor02. The bids value submitted was 150 and 110, respectively.

It is expected that the investor@example.com acquires the ownership of the asset Asset01, and his balance reduced to the value of fifty (50). Also, that sme01@example.com increment his balance value to one hundred and fifty (150) and his auction transit to the state *"Sold"*.

Figure 35 is a packet capture when requested the transaction *TxCloseAuction*, and it shows five red rectangles with some information in order to highlight the process of ownership transfer. Also, this request will provoke a chain reaction throughout the Hyperledger containers as we will show above.

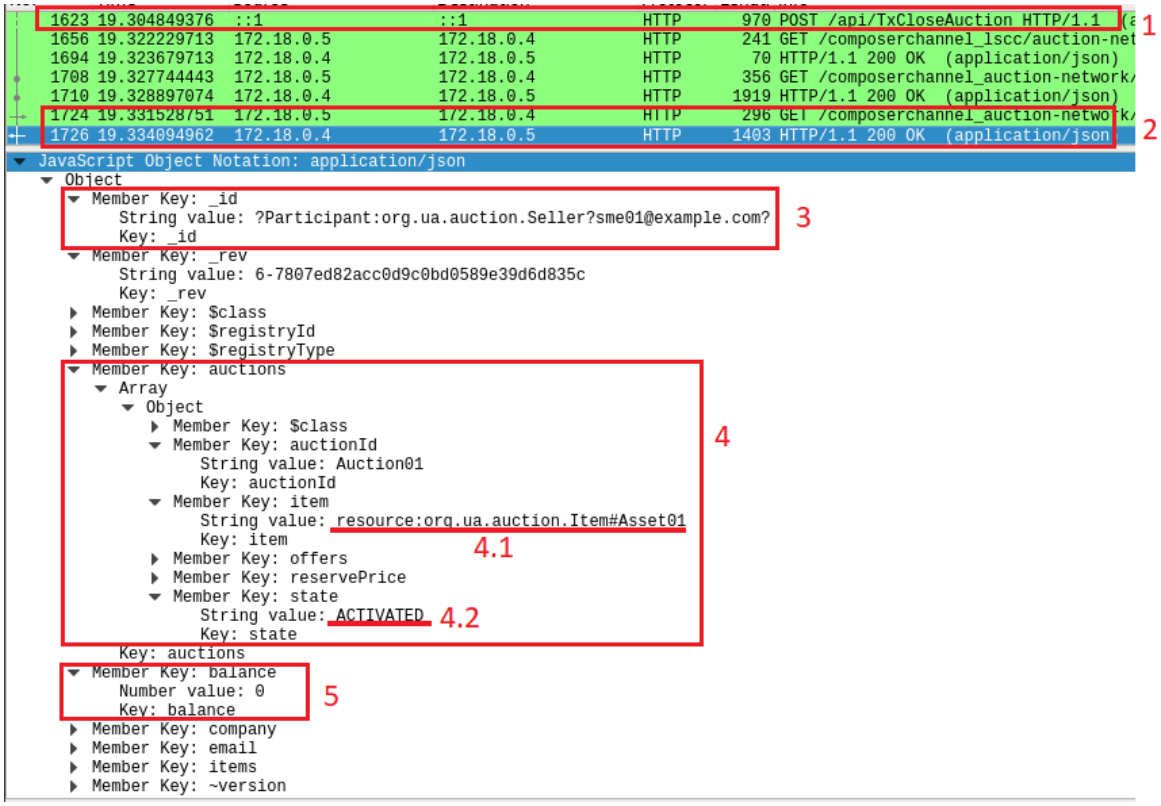

Figure 35 – Packet Capture evinced the SME state before transaction TxCloseAuction.

The request was sent by an entity with the identification: *sme01@example.com*. The red rectangle identified with the number 1, is the HTTP request that orders the execution of the transactions *TxCloseAuction*. However, before the transaction performed. Some information is retrieved from the Blockchain Service. The rectangle identified with the number 2 contains one request and the respective response that retrieves that information. It is requested by the fabric-peer container and responded by fabric-ca. The requests send are authenticated; thus they are successfully executed.

The red rectangles 3, 4 and 5 surround some information from the response, namely: the participant class type and the identifier on red rectangle 3; the auctions been close, which is identified by the attribute "*auctionId"*. Also note that the item on auction and the state are highlight and mark as 4.1 and 4.2 respectively; the balance to this participant has the value 0 (zero) at that current time.

The transaction takes place into the fabric-peer, and a new block is created. Figure 36 shows the fabric-peer container send a HTTP request with the method PUT, surrounded by a red rectangle and identified by the number 1.

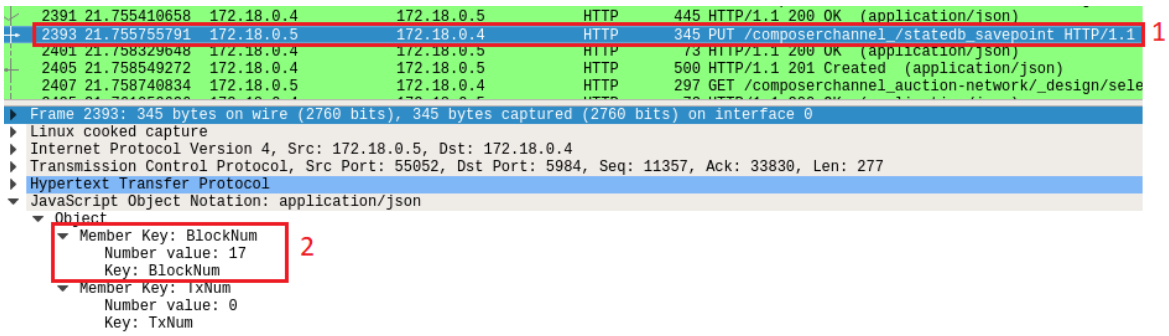

Figure 36 - New block created.

The red rectangle identified by the number 2 is the information sent, and as we can observe the new block created is block number 17.

Follow by the response to the client web application, demonstrated in Figure 37, which shows the HTTP response surrounded by the red rectangle identified by the number 1.

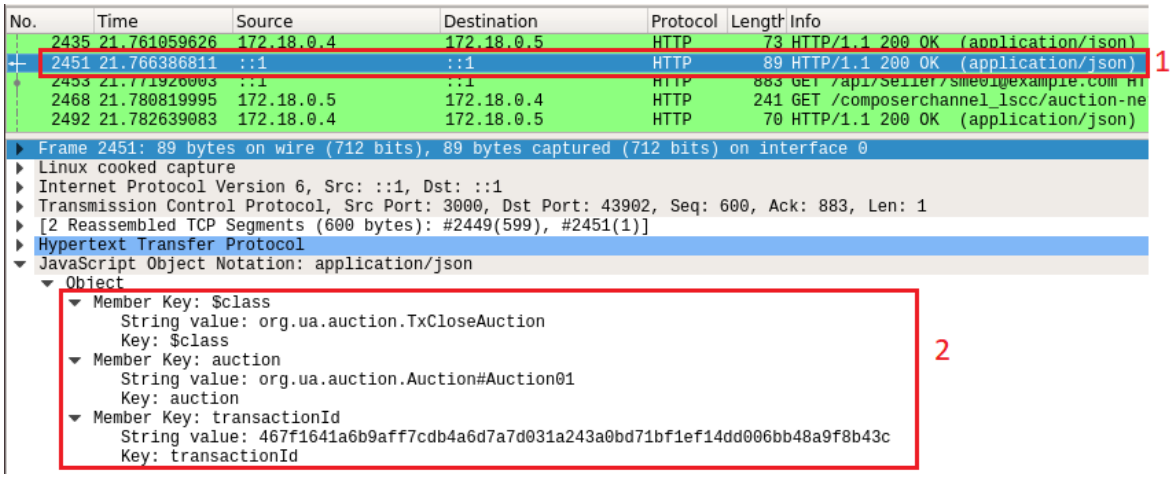

Figure 37 - Feedback from transaction TxCloseAuction completed.

Attach to this response, is the information about the transaction executed, which is surrounded by the red rectangle identified by the number 2. We can obverse the class type of the transaction executed, this case was *"org.ua.auction. TxCloseAuction"*. Also the class type and the identification

of the asset modified, this case was the *"org.ua.auction.Auction"* as the class type and *"Auction01"* as an identifier, the third is *transactionId* which means the transaction identifier, value generated after its execution.

Figure 38 shows the final state for the participant who created the auction, the SME with the identifier, sme01@example.com. The figure contains three red rectangles, the rectangle identified with the number 1 is the response for the fabric-ca to the fabric-peer, which contains the pieces of information presented in the rectangles identified by the number 2, 3 and 4.

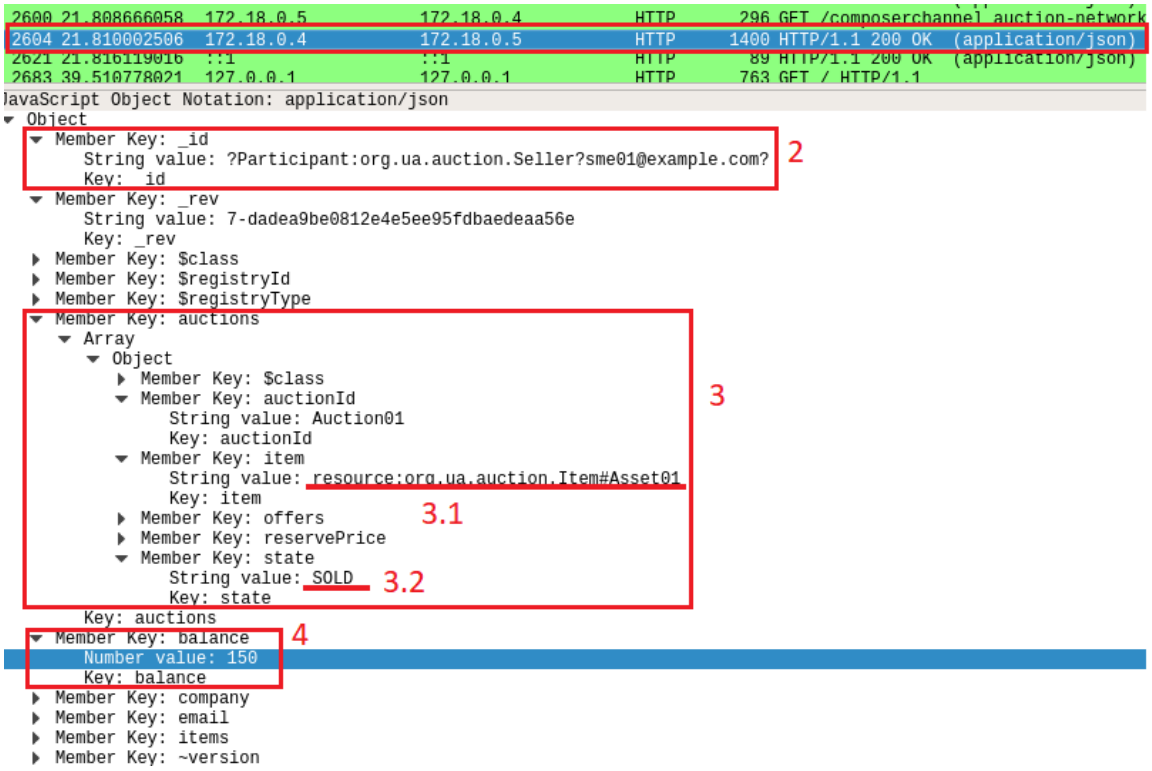

Figure 38 - SME new state after the transaction TxCloseAuction.

The rectangle 2 contains the identifier expected as rectangle 3 and the highline 3.1, but now observe that the Highline 3.2, it changed the value to *SOLD* as expected geving the scenario in the beginning of this section. Furthermore, note that the balance for this identity has now incremented for the expected value of one hundred and fifty (150).

Figure 39 shows the state of the participant that won the auction, retrieved by the HTTP response surrounded by the red rectangle identified by the number 1. As expected, it was the investor with the identification investor02@example.com.

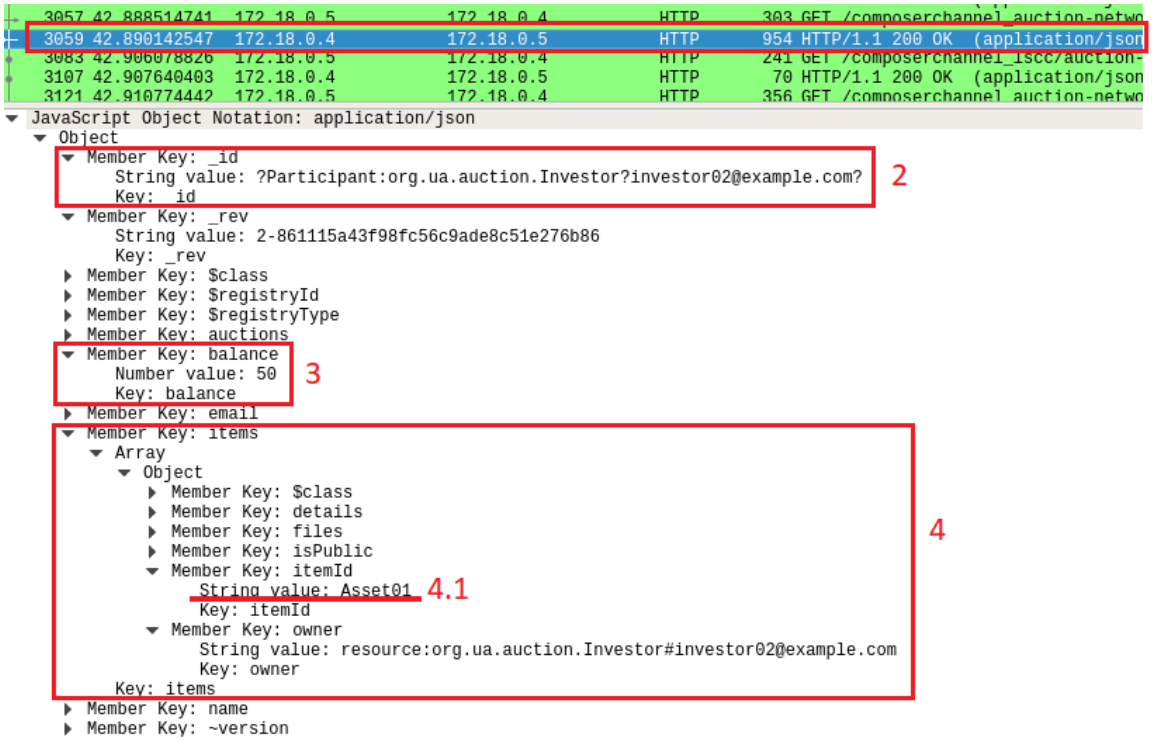

Figure 39 – Investor new state after having acquired the Asset01.

The red rectangles with the identified by the number 2 confirm the participant identification, the rectangle identified by number 3 is the value resulted from the initial value minus the bid value placed. The rectangle identified by the number 4 is the asset that the participant now owns.

6.4. Transactions

As mentioned throughout this thesis, one of the goals is data transparency, integrity, credible data. This is possible when peers participating on the network had at least read access to information about the occurred transactions, which are represented on the blockchain.

6.4.1. Historical Data

As mentioned in Section 3.6.2, the Hyperledger Composer framework offers SDK and API connection to Hyperledger Fabric runtime. The script showed in Figure 40 allows us to through the

Hyperledger Composer node based project to retrieve all the information into the block of the blockchain network. The retrieved data is present in the repository mention in Section 1.4 and can be used for further analyses.

```
const { inspect } = require('util');
const { BusinessNetworkConnection } = require('composer-client');
const fs = require('fs')

async function run() {
  const connection = new BusinessNetworkConnection();

  // Connect to the blockchain using admin credentials.
  // These credentials should be available in your local keystore.
  await connection.connect('admin@auction-network');

  // Native API provided through the Fabric SDK, allows much more low-level operations than Composer.
  const nativeApi = connection.getNativeAPI();

  // Connect to the channel where the transactions are happening, the default is "composerchannel".
  const channel = nativeApi.getChannel('composerchannel');

  strm = fs.createWriteStream('blockchain', {flags: 'a'}); // create file into the file system

  const index = 0;    // Stores the block counter
  const block;        // Stores the block information

  do {
      block = await channel.queryBlock(index);  // Grab a block by it's number
      data = JSON.stringify(block).toString();  // Convert block information to JSON format
      strm.write(data)                          // Write the block into the file
      index++;

  } while(block != undefined);

  strm.end();   // Close file into the file system

  await connection.disconnect();    // Close admin connection
}
run();
```

Figure 40 - Script to read blockchain on Hyperledger Fabric environment.

6.4.2. Performance

Typical online auction such as on eBay, as soon the auction is over the winner is almost instantly know. This is possible because it is a centralize auction. They define the rules and the system used by them as to be trusted by the client who used it. Delays to show the auction winner may occur. However, they are not frequent and are solved in least than 24h according to their customer services.

An example of permissionless blockchain, it is the Bitcoin network, which typically the transaction needs six blocks to be considered confirmed transactions. Each block on average is

created every 10 minutes. Thereby a transaction on the bitcoin network needs one hour to be confirmed.

As mentioned in Section 3.5, consortium blockchain networks are faster than public blockchain networks. This happens because more decentralization implies more performance to reach consensus. Thereby we can guaranty that our solution can confirm transactions made in less than one hour, but also it highly depends on the underlying consensus algorithm used to reach consensus.

Throughout our tests, into the packets captured through Wireshark we can observe on the green background area, on the second column, the time passed after start the capture.

For instance, Figure 33 shows complete packets exchanged capture throughout our solution. We can observe that the HTTP request to start the transaction *TxAddAuction* was invoked after 2.2256 seconds from the capture started time. The HTTP request to save the transaction made into our API Relation Service was requested on second 4.6971 after the capture started time. Thus we can conclude that between the transaction request and its completion was 2.4715 seconds.

Another example is the example demonstrated throughout Section 6.3.3 when transferring the asset ownership. The action show at 19.3058, after the start the packet capture started, in Figure 35 mark as the red rectangle number 1, represent the HTTP request to start the process to close the auction. Soon after at 21.7663 in Figure 37 also show on red rectangle number 1 show the end of the transaction with the respective transaction identification. Thus, we conclude that between the transaction request and its completion was 2.4605 seconds.

However, out captures were done on localhost, which means that if done separate machines exist more overhead on the HTTP request to the blockchain network as well into the blockchain network. Nevertheless, we are confidante that is possible to reach consensus and determinate the auction's winner in less than a business day.

7. Conclusions

Throughout this thesis, we got high and low points. Both with a great lesson to be applied to future projects. Initially, the engagement for this thesis was with high enthusiasms. Not only because it is a trending theme for the last several months, but also because throughout the academic journey, this technology was not addressed. Thus, we felt the need to fulfil this gap and get out of the conform zone.

Furthermore, this theme goes beyond the technology context. It is highly connected to the financial services, which on its own, it is a very interest context. We gain much financial maturity because of its context, manly in the debt-based assets that our research provided. Also, we discover a new type of services, the P2P Lending platforms. They seem to be a good investment alternative that we will continue to explore outside of this project.

A pitfall this context provided, it was about time management. We enrol to much on the financial activity leading to work that ultimately, it is not presented into this thesis.

The most unknown and main framework used was the Hyperledger Composer. We find this framework to be a high starting point. However, if the goal is to develop and explorer the full capacity of the blockchain network, the Hyperledger Composer does not provide that deep level of understanding to reach that goal. Instead, it provides tools to build other services and develop other technologies on top of the blockchain network.

Right after we went through some tutorials about the Hyperledger Composer, we should have decided to go for the Hyperledger Fabric framework. The focus should be on the data persistence on distributed environments. Not on two different web servers with two different types of persistence alongside with a web application, plus the blockchain network. However, the results show adaptability and knowledge of several technologies and frameworks.

Nevertheless, since the scope of this thesis was the implementation of blockchain into an auction management system. We find that the benefits of the use of blockchain technology into auction pass through the credibility offered by the network. Also, the traceability and immutability implicit at its core, it allows the auction system to have significant integrity that may be useful to several uses cases. Such as the use for a public contract, or only to promote a fair auction between several different parties, in which the participant may join the network and verify itself the transaction made for an auction.

We concluded by saying that the understanding that this technology is still immature and small design decision has a strong influence on his use and behaviour. In a way, everybody is still looking for the most suitable blockchain architecture for a business solution. Moreover, we find that Hyperledger Fabric as the potential to fulfil that need.

On a final note, we learn a lot about blockchain technology which adds excellent value about this new trend for a future professional career. Learning about the mechanism needed to make it function correctly, the different consensus algorithms and the read of several types of blockchain papers was very rewarding. As well the time spent writing this thesis because English is not our native language, this thesis was an intensive training to be able to communicate in the future, if an international opportunity shows up.

7.1. Future Work

This work contains a client web application, which is not fully tested for the interaction with all the available API's. To further support the auctions interaction in a secure matter, we propose the development of a well-structured application web.

Also, the authentication used was by the GitHub OAuth2.0 service. On a next reimplementation should be considered in apply and e-gov IdP for the user authentication.

At the beginning of this year, Hyperledger collaborate group start a new project call Hyperledger Explorer[87] a web-based tool that invokes, deploy and query blocks throughout the blockchain network. This application promises to be a useful tool to support a reimplementation of the blockchain network if needed.

On a production environment, an application with the same characteristics as our solution is capable of generating a large amount of data. Which on they own, it is a valued asset, it may provide a statistically or ranks analysis of assets and consumer behaviour for better financial decision.

“The successful warrior is the average man, with laser-like focus.”

- Bruce Lee

Appendix A

European case study: UK and Belgian, two polar cases

A Master's Thesis, "Regulation of European peer-to-peer lending Fintechs" (2017) by Clara Naïdji[88], it discusses regulatory frameworks apply by European countries, namely in United Kingdom (UK) and Belgium, which we will review in this section and analysis of how each country applies regulation differently. This document also recommends as further reading about European P2P Lending platforms.

UK: Regulatory sandbox

Within Europe, it is estimated that 80% of the P2P Lending loans emitted it was in the UK alone. This is due to the fact it was the birthplace in 2005 of the first service of this kind, the platform Zopa[89]. Since then have lent over 2 billion £ to consumer resident in the UK. Later in 2010, was created Funding Circle[90], other P2P Lending platforms also have been lending over 2 billion £ but with clients spread in Europe and specialize in B2B operations.

The success of theses platform largely relies on the sandbox concept applied. They are allowing the creation of an environment with a high level of freedom within a controlled environment to test methodologies, product or services without severe consequences to the external environment. However, they are supervised by regulators to learn and incorporate safeguards to future platforms.

These two platforms are two among keys actors' group in UK P2P Lending industry because they were the firsts one to create a fund to cover losses to non-performing loans. They founder Peer-to-Peer Finance Association (P2PFA)[91], a self-regulating organisation that aims to protect P2P consumers and share the best practices. At this moment, this association represent more than three-fourths of the total volume of P2P loans in the UK and already set standards for its members to increase transparency.

Furthermore, the UK government showed support on this activity by lending 20 million £ in 2012 and 40 million in 2014 via the platform Funding Circle. "The UK government estimates that through a multiplier effect, the 40 million £ lent to Funding Circle resulted in 450 million £ lent to UK SME and that the 20 million £ loans resulted in 130 million £ loans made to SMEs (UK Government,2014)."

Regulators, however, fear that investor is not educated about the risks involved and the difference between equity and lending crowdfunding as well as the fear that P2P Lending platforms are not fully transparent on advertisements about, the risk of default.

Belgium: Ban

The same paper states “Debt-based crowdfunding, which refers to peer-to-peer lending, is forbidden by law in Belgium.” However, one platform was authorized in 2017, called Mozzeno[92]. As a consequence, it has to modify the business model to fit with regulations and had to wait over a year for gathering all licenses. It specializes in consumer lending, and it reports that have received over 200 loan application in which 138 investors financed eight. The methodologies used are very different among other European platforms. For example, investors and borrowers have no formal relationship, and there is no bank drafting the loan agreement.

Moreover, similar to how bonds are an issue, the Mozzeno securitizes the loans, and the investors buy these securities. “Therefore, the Belgian Financial Services and Markets Authority authorized Mozzeno to perform their activities by classifying it as a notes issuer”.

However, two international platforms find a way to get authorization to operate in Belgium. It is the case of Bondora[93] and Mintos[94] but the volumes they operate are unknown. They specialized in low-risk SMEs, and the time this document is being written, the market for SMEs in native Belgium platform is non-existent.

Being forbidden by law creates a financial stigma on these platforms. At the same time, limit the financial engineer and creativity to innovate with new product or services. SME’s to find capital rely more on loan from family, friends and banks as well. According to the European Commission in 2016, Belgium has one of the lowest default loan rates, 3,5% of the total to be precise. Since to evaluate a loan is SMEs pass thought a restrict process. When requesting a sample of refused loans, the report presents that 15% of SMEs requesting loans were rejected in 2015. Which 40% for do not provide sufficient guarantees and one third for do not invest more personal capital. However, it is shown that the Belgium government support one out of ten SMEs through public investments.

Appendix B

P2P Financial Services: LendingClub

Between some alternatives, we analysis a platform LendingClub[95]an American P2P Lending service leading the market in this context. Instead of lending a significant amount to a single entity, enables you to lend the small amount to a range of different people or SME's, this way if one of those peoples does not pay back the interest in the overall portfolio can cover the lost.

The platform LendingClub, as a single person, a hypothetical borrower can take the best interest rate, is an above 6.4% to a below 9% annually. These rates are given assuming that exist stable income, low debt in the individual record. In the other end, assuming that exists huge debt and terrible personal record with a high risk of default, it can range between 28.55% to 30.99%. These loans range from 36 to 60 months (3-5-year term). A detailed table can be found on its own website[96].

For an investor, it charges a 1% fee on any payments you receive from the borrower, for whatever return the investor was getting. Which can be considered high since, at the end of the borrow, the interest rate is at least 6.4%. For borrowers, exist several fees for different situations that may or not occur during the process. Unsuccessful payment and late payment fee are self-explanatory types. Check processing fee is applied when payments are processed via checks. Nevertheless, these types of service must maintain sustainability.

For a possible borrower, the process of listing for a loan for business contains several steps. It starts with a form that includes minimal personal data, general information about the business being more details about financial information. At the end of the process, some platforms such as LeadingClub are capable of processing very raptly if they are or not willing to advance with the capital in what interest rate. In some others, the request is posted on the website in order to attract investors. Almost every platform has a protocol with banks where it will be used to deposit the loan and withdraw monthly payments. For the investor, the process is more straightforward than the borrow. After been authenticated with personal and general financial data, it can explore the listing request and find a suitable application. The choice of the amount invested ranges from dozens to several thousands of dollars. The investment process is made either by posted price or auction. The most common is use posted price to fulfil the amount request from several investors. The auctions are more used when an investor already possess allocation in investment and needs to sell that allocation for liquidity proposes.

Typically, the consumers on these platforms do not acquire service from traditional banks. The reasons may range from not having the credit requirements, the suitable ratio from income and debt or do not want to deal with a lengthy and slow verification process. Moreover, many P2P financial platforms assume no risk in analysing the creditworthiness of the borrowers. For example, in the case of Lending is noted on the license and agreement "member loans are made without obtaining any documentation of the borrower applicant's ability to afford the loan". Furthermore, in the section Credit Decisioning and Scoring Process "Although we may verify a borrower applicant's income, our underwriting and credit decisions are based on stated income." Moreover, this seems like people can easily take advantage of. In practical term, the borrower's debt is sold to third-parties debt collection agencies. As stated, "Our internal servicing team and professional third-party debt collection agencies collect payments from delinquent borrowers in compliance with the extensive consumer protection laws related to servicing and collections activities."[97] Which means the investor gets paid back whatever the LendingClub has received with the selling minus LendingClub's net fees, which generally are cent in dollar.

Lastly, taxation also must be considered. Assuming that everything runs well, and the investor receives the initial capital, and interest rates, this type of income is taxed as ordinary income**.** Which means that has one of the highest marginal tax rates in IRS, 28% for Portugal. For small and even some medium terms invest, provable does not make lose liquidity compared to long term investments.

[1] P. Schueffel, “Taming the Beast: A Scientific Definition of Fintech,” *J. Innov. Manag.*, vol. 4, no. 4, pp. 32–54, 2016.

[2] Accenture, “Global Fintech Investment Growth Continues in 2016 Driven by Europe and Asia, Accenture Study Finds,” 2016.

[3] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *J. Gen. Philos. Sci.*, vol. 39, no. 1, pp. 53–67, 2008.

[4] “Royal Flora Holland,” 2019. [Online]. Available: https://www.royalfloraholland.com/en. [Accessed: 25-Feb-2019].

[5] J. Kagan, “Peer-to-Peer Lending Definition.” [Online]. Available: https://www.investopedia.com/terms/p/peer-to-peer-lending.asp. [Accessed: 03-Apr-2019].

[6] D. Drescher, *Blockchain basics: A non-technical introduction in 25 steps*. 2017.

[7] “Investopedia.” [Online]. Available: https://www.investopedia.com/. [Accessed: 29-Oct-2019].

[8] A. Barone, “Asset,” *Updated Mar 6, 2019*. [Online]. Available: https://www.investopedia.com/terms/a/asset.asp. [Accessed: 08-Mar-2019].

[9] J. Chen, “Commercial Paper.” [Online]. Available: https://www.investopedia.com/terms/c/commercialpaper.asp.

[10] J. Kagan, “Loan Definition,” 2019. [Online]. Available: https://www.investopedia.com/terms/l/loan.asp. [Accessed: 26-Mar-2019].

[11] A. Farley, “Loan Note Definition,” 2019. [Online]. Available: https://www.investopedia.com/terms/l/loan-note.asp. [Accessed: 26-Mar-2019].

[12] J. Kagan, “Interest Rate.” [Online]. Available: https://www.investopedia.com/terms/i/interestrate.asp. [Accessed: 08-Mar-2019].

[13] W. Kenton, “Credit.” [Online]. Available: https://www.investopedia.com/terms/c/credit.asp. [Accessed: 08-Mar-2019].

[14] “Liquidity Definition.” [Online]. Available: https://www.myaccountingcourse.com/accounting-dictionary/liquidity. [Accessed: 15-Mar-2019].

[15] W. Kenton, “Current Ratio.” [Online]. Available: https://www.investopedia.com/terms/c/currentratio.asp. [Accessed: 15-Mar-2019].

[16] “Financial Instruments.” [Online]. Available: https://thismatter.com/money/banking/financial-instruments.htm. [Accessed: 30-Oct-2019].

[17] G. Edinburgh, “Growing the global economy through SMEs Contents,” *Grow. Glob. Econ. through SME’s*, vol. 1, no. 1, pp. 1–44, 2013.

[18] A. Hayes, “FICO Score.” [Online]. Available: https://www.investopedia.com/terms/f/ficoscore.asp. [Accessed: 03-Apr-2019].

[19] C. B. Murphy, “Debt-to-Income Ratio,” *Updated Feb 12, 2019*. [Online]. Available: https://www.investopedia.com/terms/d/dti.asp. [Accessed: 20-Mar-2019].

[20] J. Kagan, “FICO,” *Updated Mar 6, 2018*. [Online]. Available: https://www.investopedia.com/terms/f/fico-fair-isaac.asp. [Accessed: 20-Mar-2019].

[21] “Unsecured Loan.” [Online]. Available: https://www.investopedia.com/terms/u/unsecuredloan.asp. [Accessed: 29-Oct-2019].

[22] W. Kenton, “Lien,” *Updated Jun 12, 2018*. [Online]. Available: https://www.investopedia.com/terms/l/lien.asp. [Accessed: 20-Mar-2019].

[23] K. Greiner, “How Much Student Loan Debt Is Too Much?,” *J. Student Financ. Aid*, vol. 26, no. 1, 1996.

[24] “The 2007-08 Financial Crisis in Review.” [Online]. Available: https://www.investopedia.com/articles/economics/09/financial-crisis-review.asp. [Accessed: 30-Oct-2019].

[25] Financial Consumer Agency of Canada, “Payday Loans: Market Trends,” 2016.

[26] “Financial Consumer Agency of Canada.” [Online]. Available: https://www.canada.ca/en/financial-consumer-agency.html. [Accessed: 05-Apr-2019].

[27] N. Raymond, “Godfather of payday lending,” *2018*. [Online]. Available: https://www.reuters.com/article/us-usa-paydaylending-crime/godfather-of-payday-lending-sentenced-to-14-years-in-u-s-prison-idUSKBN1JW2XH. [Accessed: 29-Mar-2019].

[28] CGAP, “Commercial Loan Agreements,” 2006.

[29] R. M. Nash and E. Beardsley, “The rise of the new Shadow Bank,” *Futur. Financ.*, no. January, p. 69, 2015.

[30] “Godman Sachs.” [Online]. Available: https://www.goldmansachs.com/. [Accessed: 30-Oct-2019].

[31] H. Tang, “Peer-to-Peer Lenders Versus Banks: Substitutes or Complements?,” *Rev. Financ. Stud.*, vol. 32, no. 5, pp. 1900–1938, 2019.

[32] “Thebanks.” [Online]. Available: https://thebanks.eu/. [Accessed: 30-Oct-2019].

[33] D. Lucking-Reiley, “Auctions on the Internet: What’s Being Auctioned, and How?,” *J. Ind. Econ.*, vol. 48, no. 3, pp. 227–252, 2003.

[34] Amazon, “Amazon auctions.” [Online]. Available: https://a2z.bstock.com/. [Accessed: 07-Oct-2019].

[35] Docapesca, “Docapesca Auctions.” [Online]. Available: http://www.docapesca.pt/pt/leiloes-online/pescado-do-mar.html. [Accessed: 19-Oct-2019].

[36] M. Colier, *Starting an eBay Business For Dummies*, vol. 53, no. 9. 2018.

[37] “Auctionity.” [Online]. Available: https://www.auctionity.com/. [Accessed: 15-Jun-2019].

[38] “DomRaider Group.” [Online]. Available: https://www.scmagazine.cz/casopis/04-16-04-16/globalizace-mesta-migrace-rozhovor-se-saskii-sassen_locale_cs/#domraider. [Accessed: 06-Jun-2019].

[39] “Ethereum.” [Online]. Available: https://ethereum.org/. [Accessed: 21-Oct-2019].

[40] K. A. Manchester, “A Beginner’s Guide,” *Gen. Music Today*, vol. 15, no. 3, pp. 8–12, 2002.

[41] “Double-Spending Problem.” [Online]. Available: https://en.wikipedia.org/wiki/Double-spending. [Accessed: 15-Oct-2019].

[42] Microsoft, “Understanding Public Key Cryptography,” 2014. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/tn-archive/aa998077(v=exchg.65)?redirectedfrom=MSDN. [Accessed: 24-Oct-2019].

[43] D. Cynthia and N. Moni, “Pricing via Processing or Combating Junk Mail,” 1385.

[44] M. Jakobsson and A. Juels, “Proof of Work and Bread Pudding Protocols.”

[45] J. Siim, “BlackCoin’s Proof-of-Stake Protocol v2,” pp. 1–9, 2017.

[46] A. Poelstra, “On Stake and Consensus,” pp. 1–12, 2015.

[47] V. Arasev, “Proof of Authority.” [Online]. Available: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper. [Accessed: 20-Oct-2019].

[48] Nick Szabo, “Smart Contracts,” 1994. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwint

erschool2006/szabo.best.vwh.net/smart.contracts.html. [Accessed: 20-Oct-2019].

[49] S. Voshmgir, *Token Economy: How Blockchains and Smart Contracts Revolutionize the Economy*. 2019.

[50] "XRP." [Online]. Available: https://www.ripple.com/xrp/. [Accessed: 21-Oct-2019].

[51] Block.one, "EOS." [Online]. Available: https://eos.io/. [Accessed: 21-Oct-2019].

[52] S. Foundation, "Stellar." [Online]. Available: https://www.stellar.org/. [Accessed: 21-Oct-2019].

[53] C. Inc., "Corda." [Online]. Available: https://www.r3.com/platform/. [Accessed: 22-Oct-2019].

[54] "Hyperledger Composer Architecture." [Online]. Available: https://hyperledger.github.io/composer/v0.19/introduction/solution-architecture. [Accessed: 21-Oct-2019].

[55] "The Linux Foundation." [Online]. Available: https://www.linuxfoundation.org/. [Accessed: 21-Oct-2019].

[56] "The Linux Foundation Membership." [Online]. Available: https://www.linuxfoundation.org/membership/. [Accessed: 21-Oct-2019].

[57] V. Buterin, "Ethereum White Paper," *Etherum,* no. January, pp. 1–36, 2014.

[58] "Ethereum Get Started," 2019. [Online]. Available: https://ethereum.org/developers/#getting-started. [Accessed: 21-Oct-2019].

[59] "Hyperledger Wiki." [Online]. Available: https://wiki.hyperledger.org/. [Accessed: 21-Oct-2019].

[60] T. L. Foundation, "Hyperledger Fabric." [Online]. Available: https://www.hyperledger.org/projects/fabric. [Accessed: 10-Aug-2019].

[61] T. L. Foundation, "Hyperledger Composer." [Online]. Available: https://www.hyperledger.org/projects/composer. [Accessed: 10-Aug-2019].

[62] C. Cachin, "Architecture of the Hyperledger Blockchain," 2016. [Online]. Available: https://www.slideshare.net/ormium/architecture-of-the-hyperledger-blockchain-fabric-christian-cachin-ibm-research-zurich.

[63] "Yeoman." [Online]. Available: https://yeoman.io/. [Accessed: 21-Oct-2019].

[64] "Javascript." [Online]. Available: https://www.javascript.com/. [Accessed: 21-Oct-2019].

[65] “GDPR,” 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj. [Accessed: 09-Nov-2019].

[66] K. Wüst and A. Gervais, “Do you need a Blockchain?,” *IACR Cryptol. ePrint Arch.*, no. i, p. 375, 2017.

[67] O. Corporation, “MySQL’s Website.” [Online]. Available: https://www.mysql.com/. [Accessed: 02-Dec-2019].

[68] I. MongoDB, “MongoDB.” [Online]. Available: https://www.mongodb.com/. [Accessed: 02-Dec-2019].

[69] M. Jones, “The OAuth 2.0 Authorization Framework: Bearer Token Usage,” 2012. [Online]. Available: https://tools.ietf.org/html/rfc6750. [Accessed: 30-Oct-2019].

[70] I. Wikimedia Foundation, “Ubuntu.” [Online]. Available: https://en.wikipedia.org/wiki/Ubuntu. [Accessed: 30-Oct-2019].

[71] I. Wikimedia Foundation, “Debian Website.” [Online]. Available: https://en.wikipedia.org/wiki/Debian. [Accessed: 30-Oct-2019].

[72] O. Corporation, “Workbench.” [Online]. Available: https://www.mysql.com/products/workbench/.

[73] T. A. S. Foundation, “Netbeans’ Website.” [Online]. Available: https://netbeans.org/. [Accessed: 02-Dec-2019].

[74] Pallets, “Flask.” [Online]. Available: http://flask.palletsprojects.com/en/1.1.x/. [Accessed: 02-Dec-2019].

[75] Microsoft, “VSCode’s Website.” [Online]. Available: https://code.visualstudio.com/. [Accessed: 02-Dec-2019].

[76] D. Inc., “Docker.” [Online]. Available: https://www.docker.com/. [Accessed: 02-Dec-2019].

[77] Google, “Angular’s Website.” [Online]. Available: https://angular.io/. [Accessed: 02-Dec-2019].

[78] E. Andersson, Z. Berggren, E. Andersson, and Z. Berggren, “A Comparison Between MongoDB and MySQL Document Store Considering Performance,” p. Angeles, L., Advocacy, S., Location, O. (2002)., 2017.

[79] M. Rouse, “Sharding Definition,” 2011. [Online]. Available: https://searchoracle.techtarget.com/definition/sharding. [Accessed: 30-Oct-2019].

[80] “JSON.” [Online]. Available: https://www.json.org/json-en.html. [Accessed: 02-Dec-2019].

[81] H. Composer, “Using Google OAUTH2.0 with a REST server,” 2018. [Online]. Available: https://hyperledger.github.io/composer/v0.19/tutorials/google_oauth2_rest. [Accessed: 30-Oct-2019].

[82] Bootstrap, “Bootstrap Website.” [Online]. Available: https://getbootstrap.com/. [Accessed: 30-Oct-2019].

[83] S. Software, “Swagger Website.” [Online]. Available: https://swagger.io/. [Accessed: 02-Dec-2019].

[84] W. Foundation, “Wireshark’s Website.” [Online]. Available: https://www.wireshark.org/. [Accessed: 02-Dec-2019].

[85] I. E. T. Force, “RFC 7231.” [Online]. Available: https://tools.ietf.org/html/rfc7231. [Accessed: 02-Dec-2019].

[86] M. Corporation’s, “Mozilla Firefox.” [Online]. Available: https://www.mozilla.org/en-US/firefox/. [Accessed: 02-Dec-2019].

[87] T. L. Foundation, “Hyperledger Explorer.” [Online]. Available: https://www.hyperledger.org/projects/explorer. [Accessed: 02-Dec-2019].

[88] C. Naidji, “Regulation of European peer-to-peer lending Fintechs Regulatory framework to improve SME’s access to capital.” 2017.

[89] Z. B. Limited, “Zopa.” [Online]. Available: https://www.zopa.com/.

[90] F. C. Limited, “Funding Circle.” [Online]. Available: https://www.fundingcircle.com/uk/.

[91] P. 2018, “P2PFA.”

[92] mozzeno services NV, “Mozzeno.” [Online]. Available: https://www.mozzeno.com/nl/. [Accessed: 02-Dec-2019].

[93] “Bondora.” [Online]. Available: https://www.bondora.com/en. [Accessed: 02-Dec-2019].

[94] A. M. Marketplace, “Mintos.” [Online]. Available: https://www.mintos.com/en/. [Accessed: 02-Dec-2019].

[95] L. Corporation, “LendingClub.” [Online]. Available: https://www.lendingclub.com/. [Accessed: 02-Dec-2019].

[96] L. Corporation, “LendingClub Tax and Fees.” [Online]. Available: https://www.lendingclub.com/investing/investor-education/interest-rates-and-fees.

[Accessed: 02-Dec-2019].

[97] L. Corporation, “LendingClub Delinquent Borrowers.” [Online]. Available: https://help.lendingclub.com/hc/en-us/articles/215483768-What-tools-does-LendingClub-have-to-deal-with-delinquent-borrowers-. [Accessed: 02-Dec-2019].