# Projectcoin: A decentralized project exchange network

6155ELE Engineering Project Report


Name: **Emilio Tereñes Castelao**

Supervisor: **Mr. Clifford Mayhew**


Programme: **Erasmus**

Date: 8/04/2018


ljmu.ac.uk

**School of Engineering, Technology and Maritime Operations**
James Parsons Building, Byrom Street, Liverpool, L3 3AF, UK.

# Contents

# List of figures

# Resumen del Trabajo de Fin de Máster

Este trabajo se basa en el desarrollo teórico de una plataforma descentralizada basada en la tecnología Blockchain. En él se exploran y analizan las diferentes opciones que existen para el desarrollo de la plataforma, así como su utilidad y su encaje en contexto actual de financiación y desarrollo de plataformas descentralizadas.

Project Network es una aplicación descentralizada, construida sobre la red de Ethereum con su propio token como forma de pago, Projectcoin. El objetivo de la aplicación es ofrecer un espacio para el encargo y realización de proyectos online. Estos proyectos conectan a clientes con creadores y los vincula a través de Smart contracts. Estos contratos gobiernan el desarrollo del proyecto, en ellos se pueden establecer objetivos, plazos, resultados y recompensa económica. Estos contratos pueden recurrir a un arbitraje de terceras personas en caso de disputas sobre la validez del resultado obtenido, teniendo en cuenta los objetivos marcados. La plataforma ofrecerá un sistema integrado para todas estas acciones, incluyendo el contacto entre creadores/clientes, el intercambio de archivos recurriendo a un sistema P2P, la validación de esos datos mediante el almacenaje de los hashes de los archivos en la blockchain de Ethereum y la creación de Smart contracts vinculantes.

Respecto a una plataforma tradicional Project Network tendría dos grandes ventajas:

- Por una parte, ofrece seguridad. Los Smart contracts gobiernan el desarrollo del proyecto, asegurando su pago y confidencialidad. Una vez estas características sean entendidas en la industria la tecnología Blockchain será ampliamente adoptada en el mundo empresarial.
- El servicio puede ser descentralizado. Los actores que intervienen en el desarrollo del proyecto buscan su propio beneficio económico. Los desarrolladores en cambio se encargan solo de velar y mejorar el ecosistema. Esto reduce el coste para los usuarios y reduce la influencia de los desarrolladores, evitando por ejemplo perdidas de datos de los usuarios.

En este proyecto se incluye además de un análisis general de la plataforma un análisis de los aspectos técnicos, cuáles son las mejores opciones y de cómo se pueden desarrollar. Estos análisis se pueden encontrar en los apartados dedicados a el desarrollo de plataformas descentralizadas en Ethereum, el almacenamiento de hashes de archivos intercambiados, el intercambio de archivos y el sistema de arbitraje.

También se incluye un análisis económico detallado del proyecto apoyado en investigaciones sobre el contexto en el que se pretende desarrollar, así como explicaciones técnicas de cómo llevarlo a cabo.

# 1. Introduction

In the Information Age, engineers are faced with the task of providing completely new solutions in almost every aspect of our lives. In a twenty years period humanity has witnessed a true revolution. Internet has converted us in the first globally interconnected society. But with twenty more years, humanity will probably witness the biggest leap forward within a generation. This project humbly aims to propose a new tool to make this advance easier.

Nowadays there is no doubt that there is a wide range of tasks that can be fulfilled and delivered in the form of a computer file. A globally interconnected society needs to take advantage of this and make possible that people from opposite corners of the world can be able to work together. Working from home has been in our predictions for the future for too much time. The propose of this project is to take this idea one step further and design a decentralized project exchange network, in which any task that can be delivered as a file, can be securely proposed, done, exchanged and paid. This project will be supported by blockchain technology, and doesn´t aim to innovate in that field, instead it explores how to adopt this new technology in the development of a decentralized project exchange platform.

Blockchain technology has emerged in the recent years as a breakthrough in the internet world. Bitcoin has been the flagship of this revolution, creating a decentralize secure payment method and grabbing the attention of the mass media. But more importantly, what Bitcoin has achieved is the emergence of many blockchain projects under his wing (1) (2). Also, blockchain technology is having a huge impact on big companies, accordingly to a recent survey (3) 60% of the companies with more than 20,000 employees are currently deploying or considering deploying it.

This project will explore and discuss the possibilities, challenges and possible solutions of creating an online project exchange platform powered by blockchain technology. The opportunities offered by this technology will be the main focus throughout this report, as they represent the biggest innovation of this idea. The introduction of this report will continue by setting the objectives, showcasing the potential uses of the platform and specifying the structure of the report in the following sections.

## 1.1. Project Aims and Objectives

The aim of this project is to design a decentralized application in which it would be safe to perform a task, in the form of a computer file, and get paid using blockchain technology. From a user perspective, the user will encounter a platform in which it will be possible to post project offers, select the preferred candidates that apply for the project and receive the project in the form of a file. All done in the same platform in a secure environment, to avoid scams or information theft. By the nature of this project was considered not to rush the development of the platform, but instead design it and translate this knowledge into the whitepaper of the project.

Every blockchain project since Bitcoin has had a whitepaper as a presentation to the community, often to look for funding. The content of a whitepaper may vary a lot, but in this case, is going to consist of three main elements, set as objectives of the project:

**1.** Evaluate the development options for a decentralized application capable of integrating smart contracts.

**2.** Identify different ways to make possible to share files in a secure way and choose the most adequate one for this application.

**3.** Set the best conditions for the usage of the platform, such as arbitration systems, fees, term conditions etc.

In addition, the report will be complimented with an economic analysis, something uncommon in other whitepapers but suitable for an engineering project.

The objectives of this project are discussed more in depth in the Plan for the project chapter, where they are fleshed out both in importance and approach.

## 1.2. Potential clients

Both as an example of how the platform intends to work and a justification of the project, three uses of the platform will be presented:

1) This platform will be an easy way to connect people form every place in the world to perform a task. Let's take the case of a natural disaster occurring in India. A newspaper like the guardian could use this platform to ask for an article and photos of the disaster, eliminating the need to send journalist to cover the news in foreign countries. Using blockchain technology also provides an easier payment method than regular international transactions.

2) Another potential use of this platform is the ability to open a project to the world. In the design of the hyperloop, Tesla run a contest to select the best design for the wagon. This type of contests are common in fields like architecture, where an initial original idea is key to the success of the project. The platform will be able to host this contests in an easy way.

3) Finally, the biggest advantage of this platform is that is a simple method of finding experts in fields that you may need to subcontract other companies for doing them. Engineering companies are faced every day with tasks that they are not prepared to do by they own. The usage of this platform will provide them a huge pool of professionals around the world, that may find the proposed task easy to deal with, and accept to do it fast and for a reasonable prize. From the employee perspective, the platform could represent an easy way to make money. As some companies may ask for task that require a lot of work, and thus a high reward, but an employee might find it an easy task because he has worked on the field before. For example, a company may ask for the design of a circuit that a just graduated student has design for his engineering project, the fulfilment of the task will require him an adaptation of his previous work.

At the end of this report a covering letter to these potential clients can be found. The propose of this letter will be to promote the platform between its potential users making use of a promotional method outlined in the economic analysis.

## 1.3. Structure

This report will introduce at first the planning of the report, reviewing how effective and accurate it was during the curse of the project, and the literature review, which shows a structured review of the resources that served for the preparation of this project. Both of this section were already presented in the interim report and had been actualized.

The next five chapters correspond to the discussion and findings of the report, these chapters conform the main body of this report. The nature of this project makes this section very important as t is where the design process of the different features of the platform is carried out and the analysis of the different technologies are explored. The last of these chapters comprises a mixture of technical information on how a decentralized application can be funded and an analysis of the prospections of the platform.

At the end of this report, to give meaning and summarize it all, there are three chapters, dedicated respectively to conclusions, future work and a potential client covering letter. In the appendixes of this report it can be found the Gantt diagram of the project, an example of a smart contract, the ER20 token protocol and a smart contract from an ICO.

All of this chapters will have its own individual introduction.

# 2. Plan for the Project

The following section aims to explain in detail the objectives of the project, also explaining their importance and projected order. Due to the restricted time, their duration must be predicted and the resources must be distributed accordingly to their importance. A Gantt diagram, result of these considerations, can be found as an appendix to this report.

## 2.1. Importance of the objectives

The project can be divided in to two main objectives. The first one is to develop the whitepaper of the project. This is the most important objective, as is it will be the core of the further development of the project. The white paper of a blockchain project consists in an explanation of the aim and uses of it, in other words, explain why it will be a useful project. This part is already outlined in the introduction of this report, but it will be further developed and rearranged once the technical details are solved. The second part of a whitepaper is a more in depth technical analysis of the project, how to achieve what has been promised, this analysis is divided in three main categories explained in the following subheadings.

The second main objective of this project is an analysis of the economics of this project. As previously stated in this report, the recent blockchain projects are funded through an ICO. This method of funding has not been studied in depth and it's interesting of being studied by its own, but in this case the project will contain an analysis centred on the success of the platform. There will be also an analysis of economic subsistence of the project, identifying potential risks and objectives necessary to sustain the development of the platform.

The following subheadings consists in a prediction of the resources needed to achieve each one of the objectives and their predicted order.

### 2.1.1. Evaluate the development options for a decentralized application capable of integrating smart contracts.

This is the most important objective in order to achieve the aim of the whole project. The platform will be developed based on blockchain technology due to the advantages that this new technology presents, previously stated in this report. The goal is to refine the outlined features of the platform, giving specific solutions to every challenge. These challenges are: being capable of doing secure transactions, integrate smart contracts in these transactions, store the file exchange in the blockchain so it is auditable and integrate an arbitration system.

#### 2.1.2. Identify different ways to make possible to share files in a secure way and choose the most adequate one for this application.

Once the previous analysis is completed, a solution for the file sharing must be found an explained in detail, it has been already proposed in this report to implement a P2P file sharing solution due to the literature research carried out. This solution has to be developed in a way that can be integrated with the blockchain technology part. The desired integration is the capability to give a user (the creator) the possibility of securely sharing the file with the other user (the client) and automatically recognize this sharing by the smart contract and store the file exchange in the blockchain.

#### 2.1.3. Set the best conditions for the usage of the platform, such as arbitration systems, fees, term conditions etc.

Once the two previous objectives are achieved, the last part of the technical analysis is to provide a clear explanation of who the platform will work. A perfect example of this is the arbitration system. In the first part has to be stated who it is possible to implement an arbitration system, but this part aims to set the conditions of how it will work. For this is necessary to have a broad picture of the project, that's why is going to be developed in parallel with the economic analysis.

#### 2.1.4. Economic analysis.

This objective is expected to be the last to complete. Although is going to be carried out in parallel to the setting of usage conditions, it will be finished once these conditions are stablished. This is the most logical way to approach this objective, as the economics behind the project must be taking in to account to set the usage conditions of the platform, but the final analysis should be carried out with a completed platform in mind.

### 2.2. Project constrains.

Time has to be considered before setting the aims and objectives of any project. In this case, it has been considered. The result of this consideration is that instead of aiming to complete the development of the platform, set the aim of this project as providing an alpha version of the platform, the aim was a detailed theoretical design of the platform.

### 2.3. Risk of successful completion.

The main risk of this project is that it success depend on providing a series of technical solutions. For example, if this platform cannot provide a way of performing smart contracts it would fail to serve to

his propose. But in the literature review of this report has been shown that there are other projects providing solutions to the challenges that this platform presents, so the risk has been greatly reduced after the completion of this research.

Another risk is that the lack of time results in an incomplete platform. This risk has been taking in to account, as a result the economic analysis of the platform is the last objective to complete, reducing the consequences of a lack of time to an unfinished economic analysis, not to an incomplete platform.

## 2.4. Critical evaluation of the plan for the project

After the completion of this project the planning has proven to be successful and for the most part an accurate prediction. It can be highlighted that the development of the project didn´t feel rushed at any time while the achievements have been satisfactory. From a personal point this has been a great improvement from my previous final project, when I felt more pressure towards the end of the project. All of the objectives, from my perspective had been met and its completion can be measured throughout this report.

One of the reasons behind this success has been the planning of the project and to force myself to fulfil objectives and write down the progress to include it in the final report. The only task that wasn´t written by the presentation of the project was the development options part. This was a conscious decision, taken to dedicate myself to learn some practical skills in programming in solidity and have a better understanding of the blockchain technology. That part was then written down in the time to integrate the final report, as there was a time margin set for that propose.

# 3. Literature review

The main goal of this section is to provide an introduction to the blockchain technology and other useful projects, the information listed here will be used in the further development of the project. This section will be divided in three main fields. Blockchain technology and development options, in which previous projects will be explained and compared, and the different options to develop decentralize platforms. Existing decentralized file sharing platforms, a review of the actual resources available to exchange files without third parties involved. And lastly, a brief note on the available research and intentions for the economic analysis of a blockchain based platform.

Other projects whitepapers are the main source of information of this literature review. There are not many research papers on blockchain technology (4), and the whitepapers are a direct source of information on how other platforms deal with problems faced in this project.

## 3.1. Blockchain projects and development options

The history of the blockchain technology is usually linked to the history of Bitcoin itself, and besides there are some earlier projects, this review will start with Bitcoin and his white paper (5), as an easy way to explain some of the concepts needed.

Bitcoin's idea was released in 2008 by Satoshi Nakamoto and Bitcoin itself was released in 2009. The goal of Bitcoin was to create an electronic payment system without the need for a trusted third party. Only in this phrase two vital things for this project are encountered: electronic payment and independence from third parties. The electronic payment method founded by bitcoin has become the main connection between all the following blockchain projects. In one way or another every one of them takes advantage of the capability to exchange electronic coins. In regards of the independence from third parties, what bitcoin achieved was to create a currency that didn't need a national bank that backed his value or any financial institution to ensure its transactions. In this project, the goal is to provide an environment free from trusted third parties and also integrate this idea of electronic payments.

On the technical side, Bitcoin solved proposed a solution for its payment method based on two pillars: signatures of the transaction and a public ledger of transaction, also called blockchain. These two solutions are based on the idea of making auditable what is private. This is achieved through mathematical functions called cryptographic hash functions, fundamental in the understanding of this project. A cryptographic hash function is a one-way function, in which an input of any given size is converted into a string of bits, called digest, that cannot be trace back to its input. But it can be always verified that a given input gives a certain digest. This makes the digest act like a public key, that can be public, and make the original value the secret key.

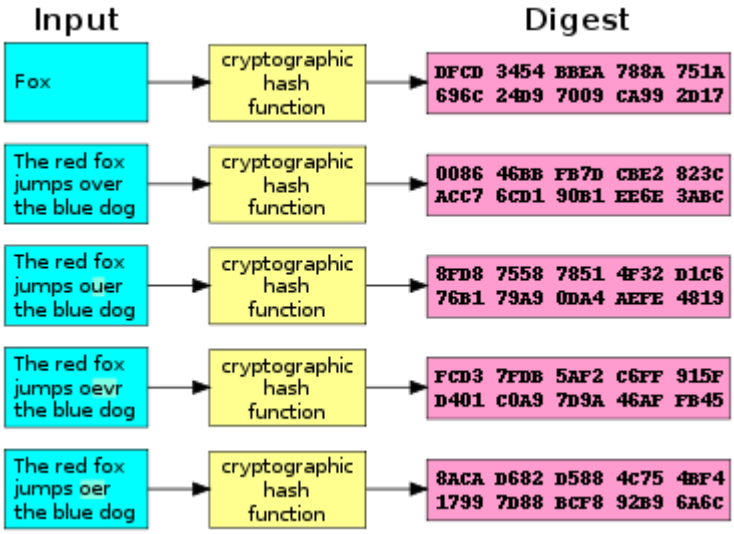



*Figure 1 Cryptographic hash function example, any changes made to the imput give a completely different digest (6)*

Every cryptographic hash function must have three safety characteristics:

1. Pre-image resistance: This property, related to the one-way function concept, means that given the hash value should be hard and time consuming to find the original message. A common example of this is the prime factorization of a big number. It is a NP problem, which makes that in practice it cannot be solved. This is commonly used in cryptography, as the product of two prime numbers can be the public key and the owner will have the secret key, the two prime factors.
2. Second pre-image resistance: Given an original value, it should be difficult to find another value with the same hash value. If this was easy, the secret key would lose its value, since it would be easy to find another one.
3. Collision resistance: It should be difficult to find two inputs with the same hash value. These pairs of numbers are called collisions, and need to be extremely uncommon to prevent any malicious use of the function.

Bitcoin uses as hashing function the SHA 256 (Secure Hash Algorithm), an evolution of the SHA 1 developed by the NSA in 1993. SHA 256 it is considered more secure than its predecessor, which has been brute forced attacked successfully in 2005, finding a collision in $2^{69}$ operations. SHA 256 operates several loops that combines the input data, divided into 512 bit blocks, with the previous result of the loop until every block has been used. In order to start the process an initial hash value is generated, this initial hash value is obtained from the fractional part of the square roots of the first eight numbers, as it has to be the same every time the algorithm is used. The combination of the data that takes place on every loop ensures that the function has the security characteristics previously defined. The complete functions used in the loops can be found online at (7), now that the principles behind the performance of the hashing function had been established, this literature review will continue with their use in the Bitcoin protocol.

Every transaction of a Bitcoin is broadcasted to the nodes on the Bitcoin network. Digital signatures were the proposed solution for the authentication of these transactions. A Bitcoin wallet has a public key (that works as an account number) and a private key (that is used to verify the transactions), that are mathematically linked. Whenever a transaction is broadcasted it is accompanied with the signature, the digest of a hash function with the transaction message and the private key as inputs. Then a special function can verify with the public key, the transaction message and the public key if the private key corresponds to this public key, and thus the transaction is valid. The mathematical concepts behind this function and how the public and private key are linked can be found here (8).

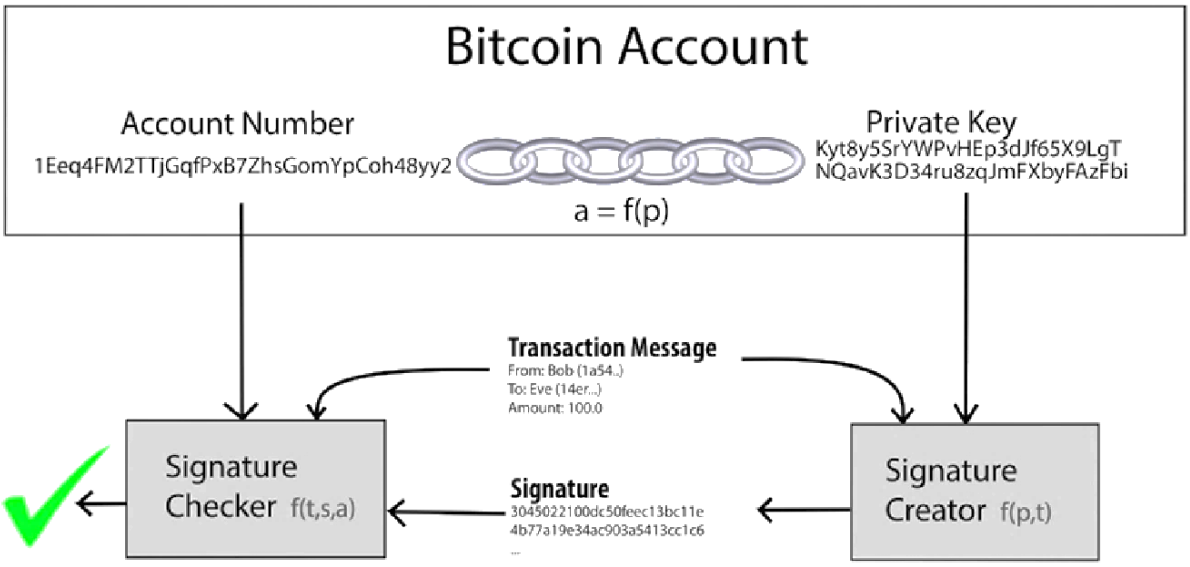



*Figure 2 Bitcoin transaction diagram (9)*

The creation of the public ledger of transaction was the other fundamental solution in the development of Bitcoin. It solves the problem of double spending, preventing someone to use its Bitcoins twice. This solution relies on the nodes commonly known as Bitcoin miners, who are the watchers of the Bitcoin network. The miners run two processes in parallel, firstly they receive the transaction messages and order them into a block and secondly, they try to solve a mathematical problem to continue with the next block. This mathematical problem links the current block with the next one, this system is called Proof-of-Work. The difficulty of the problem is adjusted to be solved on average every 10 minutes. The first node who solves it, writes the next block of transactions and its awarded with Bitcoins.

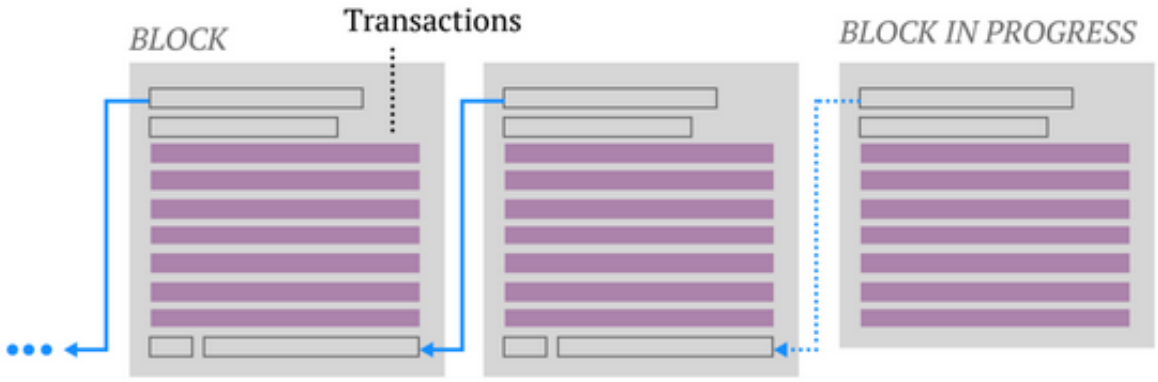



*Figure 3 Blockchain diagram (10)*

This creates a blockchain containing all the transactions, which solves the problem of double spending, as there is a record of every transaction done and its order that is accepted by all the nodes. This idea is the genius behind Bitcoin success, substituting a trusted party for many distributed auditors, and is the fundamental breakthrough of the blockchain technology.

To achieve the goal of this project is fundamental the idea of a blockchain, because it's a system that provide a reliable method to store auditable information, and this project aims to store the file exchange in the blockchain. This idea will be discussed later on this literature review while explaining Factom.

### 3.1.1. Smart contracts and Ethereum.

As stated in the book, Blockchain: Blueprint for a New Economy (11) , the blockchain technology can be divided into three stages, of which smart contracts are the second one:

> "Blockchain 1.0 is currency, the deployment of cryptocurrencies in applications related to cash, such as currency transfer, remittance, and digital payment systems. Blockchain 2.0 is contracts, the entire slate of economic, market, and financial applications using the blockchain that are more extensive than simple cash transactions: stocks, bonds, futures, loans, mortgages, titles, smart property, and smart contracts. Blockchain 3.0 is blockchain applications beyond currency, finance, and markets—particularly in the areas of government, health, science, literacy, culture, and art."

Talking about smart contracts, could be approached form the historical evolution of the concept, but for the clarity of this review it will be introduced through Ethereum (12). Smart contracts can be defined in one sentence as contracts written in code. Contracts work because if their conditions aren't respected by any of the signers it may have legal repercussions. On the other hand, smart contracts work automatically when the set conditions are met.

A football bet between two friends can be explained as a simple example of a smart contract. These two friends send the money to a neutral account controlled by the smart contract, once the result of the match

is checked the earnings of the bet will be send to the winner. None of the people involved have the capacity to interrupt the smart contract fulfilment and thus no legal binding is required. The different methods of how smart contracts check conditions in the real world and its importance to this project will be reviewed later.

Smart contracts can be performed on Bitcoin, in fact a Bitcoin transaction is a simple type of smart contract. The problem is that the language script as implemented in Bitcoin is not Turing complete. One of the biggest things missed in Bitcoin is loops, this is mainly due to avoid the halting problem, "no Turing machine can determine beforehand whether a program run in it will either terminate (halt) or run forever" (13). Overcome this issue is the main goal of Ethereum, "Ethereum attempts to do that, marrying the power of decentralized transactions with a Turing-complete contract system" (13).

This problem is solved in Ethereum by requiring Ether, the token of the platform, to do the computations in a smart contract and the amount paid has limit set on the definition of the smart contract. So in the event of malicious or bugged code, the smart contract would run out of ether before engaging the node in an infinite loop.

Another important achievement of Ethereum is the capability to create decentralize applications with its own token on top of the Ethereum Blockchain. The development of decentralize applications its done in Ethereum in two languages, Solidity and Serpent, similar to JavaScript and Python respectively, specially developed for the Ethereum platform. This topic will be later addressed with other development options.

### 3.1.2. Factom, securing data in the blockchain.

Unlike Bitcoin or Ethereum, Factom's success is yet to come. Its idea (14) is to allow a user to store hashed information in the blockchain, this means that it becomes instantly auditable (15), but prevented from theft, corruption or lost. For doing this, Factom writes the data into their data layer and afterwards that information is anchored to the Bitcoin blockchain, although is planned to anchor the information to other blockchains. To be able to write information into the blockchain the user must have entre credits, that are purchased in the token of the platform the factoids, in a similar way of how ether works for Ethereum smart contracts.

Factom is very interesting to this project, because it takes advantage of the ability to make auditable any information on the blockchain. This idea will be adapted in the development of this project, as it will offer the users the possibility to demonstrate the files exchanged in the smart contracts, a very powerful tool when dealing with intellectual property issues.

### 3.1.3. FistBlood

FirstBlood (16) is a decentralized betting platform focused on eSports. The particular aspect of this project that will be addressed here is the witness system. As stated before, in the smart contracts and Ethereum section, the smart contracts may need someone impartial to verify some external result, in this case the result of an eSports match. In FirstBlood witnesses are selected at random after sending a contribution to the smart contract address. If everything is ok two witnesses will be selected for every match and an automated application run by them will provide the result of the match. But if the result is contested a jury voting pool will be called to vote manually the result of the match, looking at the evidence provided.

FistBlood witnesses system is interesting for this project because it shows an example of easily coded witnesses system that follows the decentralize philosophy of the blockchain platforms. In this project, however a witness will not be needed in every smart contract. But once the file of the task is delivered a judging system might be needed to settle disputes. This is possible because the smart contract can hold the funds of the transaction until the two parts agree on the result. This judging system will be further developed an explained on the final report.

### 3.1.4. File sharing with blockchain technology.

Regarding file sharing through blockchain technology there are several projects headed towards it. These projects are centred on cloud storage and plan to expand their functions to allow the user to not only storage their files but to share them afterwards with another user.

Storing data directly into the blockchain is inefficient, the Bitcoin blockchain, composed only of transactions data is 95 Gb already, so a blockchain that tries to handle a big data amount is not scalable. The base of all of these projects is to provide an efficient system in which users can rent space in their hard drive in exchange for tokens. Doing so, these projects are providing storage services without central servers and a trust third party who runs them. These systems are based on two principal ideas, the data must be distributed into the hard drive of more of one user and encrypted, so the date is protected. Also for avoiding information loss the date is stored more than one time in several hosts, this also ensures the availability of the data at any time.

Could storage is a growing technology and for example Siacoin is offering an average price of 2$ per TB compared with the 23$ per TB of Amazon S3. So, these projects have a great potential of substituting the current model of cloud storage.

These projects are taken into account in this literature review because they deal with the problem of file handling in the blockchain, showing that individual people renting hard drive space is needed. So instead of attempting this, a much simpler solution is going to be presented in the second part of this literature review.

### 3.1.5. Development options.

As mentioned previously in the Ethereum section, there are several running projects that provide a development environment for new decentralize applications. These projects aim to create an easy way to develop platforms using blockchain technology. These platforms don't base its success in a breakthrough technology, but instead they build a successful idea using the available technology. An example of decentralize platform is FirstBlood (16), previously addressed on this report, that is built on Ethereum.

This project fits into the previous description, the idea behind it is not a breakthrough in blockchain technology but instead is an adaptation of the existing technologies to build a platform. As so a development platform has to be chosen for the development of the project, being Ethereum the preferred candidate at this time. However, two other possibilities can be considered if there are any inconveniences in the development with Ethereum. Lisk (17) is a platform specifically build to develop decentralize applications and bases its success in providing a more user-friendly platform than Ethereum and the ability to develop in JavaScript, a common language for developers. Neo (18) is commonly known as the Chinese Ethereum, the development options that it present differ from Ethereum in the languages used, since Neo uses common languages such as C, C#, C++ and JavaScript. But besides the advantages of the common languages used by Neo and Lisk, Ethereum is a much more consolidated platform, a factor that is much more decisive for choosing which is the more convenient.

## 3.2. Existing decentralized file sharing platforms.

As stated before the existing resources to exchange files with blockchain technologies rely on decentralized cloud storage. In the same way that mainstream file sharing platforms, such as Dropbox, rely also on cloud storage, but on centralized servers. The servers allow users to download the files at any time, but for this project is not necessary and the research was focused to find a method that eliminates the need for servers, to avoid the need of implementing a solution similar to the ones described on the previous section: File sharing with blockchain technology.

Peer to peer (P2P) has been usually associated with file sharing networks, that makes files available to any member of the network (19). The idea for this project is to emulate these networks but only connect the two users involved in the file exchange. Fortunately, this idea has been developed before by several platforms with the propose of secure and fast file sharing. Between all these platforms (20) (21) (22) Send Anywhere is probably the best one. Apart from having a browser service it has desktop applications for several operating systems and a mobile app. The objective is to integrate a similar system in this project, because is the easiest and safest way to transfer the files involved in the task performance.

## 3.3. Economics in the blockchain.

In order to talk about the economics of the blockchain technology is again necessary to address to Bitcoin. In this case to the growth of its valuation since its realise, that started at merely cents and its traded at $6.000 nowadays.

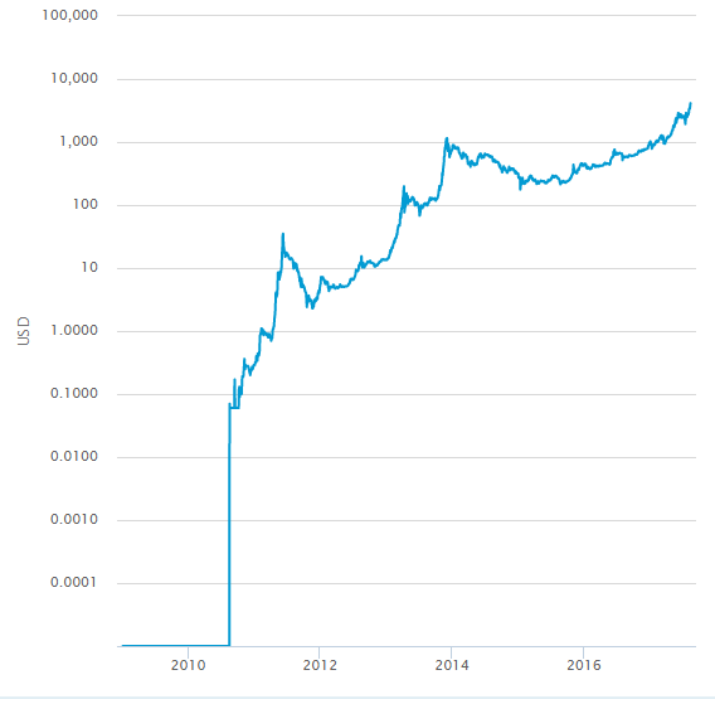



*Figure 4 Bitcoin price since inception (logarithmic scale) (23)*

This impressive growth has led to the creation of an ecosystem of different coins that together has a combined market capitalization of $165.549.529.089 where Bitcoin represents the 55.9% (24).

Every owner of cryptocurrencies can easily buy other cryptocurrencies through exchanges such us Bittrex or Poloniex. This fact, combined with the growth of the inversions in cryptocurrencies makes very easy for a new coin to grow in value, as the investors in cryptocurrencies are always looking for a new project to reinvest their profits.

Before being listed on these exchanges, new cryptocurrencies usually go through an Initial Coin Offering (ICO), where an initial amount of coins are sold to investors, working as funding for these new projects. Money raised via ICO has recently surpassed the early stage VC funding for start-ups (25). Despite its importance, and possibly due to its recent emergence, there are not many research papers about funding strategies with ICOs, instead there are some that aim to advise about investing in them (26) (27) or evaluate their trustiness (28).

The goal of this project is to design an adequate ICO for the project, identifying the strengths and risks of the ability to quickly raising money in this way. Further research will be conducted in this sense once the objectives of the ICO are established.

# 4. Development options

Although this project consists in the conceptual design of the platform, the aim is to be as specific as it can be done, and design every feature of the platform taking into account the available technology and exploring how to implement the platform into it. This headline in particular will explore the options regarding the creation of the platform and justify the use of the blockchain technology.

This headline shows the research and findings on these crucial questions on the development of the platform:

- Justify the selection of the token creation for the platform and explain the differences between the different available options.
- Choose between an existing blockchain project that allow to build the platform on it.
- Explain how a decentralized platform works and it is integrated with a user interface, and more importantly, select how this is going to be implemented in the platform.
- Show how the smart contracts works and how the basic exchange of a project can be performed through them.
- Explain how does the creation of a new token work.

## 4.1. Blockchain, token or application

As it was stated at the beginning of this report, this project doesn´t aim to develop a new technology, or take further the existing one. Instead of this, the aim of this project is to develop platform capable of integrating some of the advantages that the blockchain technology offers.

There are two big decisions to make at the beginning of the design of the platform. The first, is whether is interesting to develop an own blockchain or is better to develop the platform on top of another project. To keep this report simple, I will refer to two previous projects that were outlined in the literature research, Ethereum (12) and FirstBlood (16).

Ethereum has developed its own blockchain. Ethereum faced a big technical innovation, as it aimed to add to the existing blockchain technology a Turing complete virtual machine. Ethereum´s main developer, Vitalik Buterin, has stated several times that he tried to develop his idea inside Bitcoin but didn´t find the needed support from their community of developers, so he moved away and created his own Blockchain. Ethereum exceeded the boundaries of the existing technology, and its decision has been proved to be right, as Ethereum was created in a time where most of the coins born after Bitcoin where useless creations but now it has led a growing number of alternative projects to Bitcoin, that add real value to the Blockchain technology. On the other hand, FirstBlood was created on top of Ethereum, because it didn´t pursue a technical innovation, but it opened the eSports betting to the advantages of the Blockchain technology.

With this example it can be clear that the aim of this project is better suited with the idea of building itself on top of another project than with creating its own Blockchain. This selection has several advantages, like being part of a bigger, more recognise ecosystem of projects. But it is also a doubled edge sword, as it means that there are technical improvements that are going to be handled by the parent project, which means that the project is dependant of its success.

Once it has been decided that is better to develop the project on top of other blockchain, the second decision is whether or not is better to create a new token associated with the platform. A token created on top of another blockchain doesn´t have many different uses from the original Blockchain´s token. The only difference is that the developers can mint their own token. This means that most of the decentralized applications built on top of another token can perfectly function with their parent token. But most of them don´t choose this option, this is due to the importance of minting new tokens as this allows them to launch an Initial Coin Offering (ICO) to fund their platform. For this platform it has been selected the option of an own token. The benefits of this are outlined in the Economic analysis of this report, as an own token allows an ICO and a promotion based on the minting of new tokens, two very important parts of the economic strategy of the platform.

However, it must be stated that this decision also comes with drawbacks. Being the main that the token of the platform will be not as recognised for the public as the parent token. This is a situation affecting most of the other blockchain projects. But taking into account the great number of cryptocurrencies and their facility of interexchange, the most foreseeable future is one where this doesn´t suppose a problem, and possessing different cryptocurrencies for different uses it's a common thing.

## 4.2. Ethereum

The selected blockchain to develop the platform is Ethereum. Although other blockchains were considered, there have been three main reasons behind this decision:

- **Ethereum is currently the largest decentralized application development platform.** If we take into account the Metcalfe´s law, the effect of a telecommunication network is proportional to the square of connected users. The Ethereum ecosystem can be seen as a communication network with the propose of exchanging services and money, and having this initial advantage can be decisive in the future adoption. Being part of the biggest ecosystem can be a fundamental advantage for this project.
- **Ethereum has the longest proven working product.** This point has to be considered carefully when dealing with blockchain technology. It´s a technology based on security protocols and immutability of records, so any discovery of security flaws would be lethal to any of these platforms. In that regard, Ethereum has been exposed to such flaws for a longer time.
- **There are more online resources for developers.** Although development in Ethereum can be seen at the beginning as more challenging than others due to its unique languages of programming, the reality is that this has helped to develop an extent library of online resources. Solidity has become the de facto language of Ethereum, and thanks to the necessity of learning it, there are more specific resources dedicated to every step of the creation of a decentralized app.

Besides these advantages ultimately decided in favour of Ethereum other projects were explored as a support for this project. Here are some of them and their benefits over Ethereum.

Neo, is a Chinese blockchain project founded by Da Hongfei in 2016 (18), and has earned the name of the “Chinese Ethereum”. This has helped to increase the value of the token during 2017, following the rise of Ethereum, in the hopes that this Chinese origin helped to take over the Chinese market in the development of decentralized apps. While this is important, as it represents a huge potential growth the main advantages over Ethereum in the case of a technical analysis are others. Neo focuses its project in the smart economy. These concepts refer to the combination of Blockchain technology and the real-world economy. Two pillars of this smart economy support are the digital identity and the digital assets. The plan of Neo is to implement digital identity through the international agreed PKI (Public Key Infrastructure) X.509 standard as an essential part of their platform. Digital assets on the other hand represents the possibility to have digital ownership of an asset secured in the blockchain, allowing also their exchange. Neo also works with a Delegated Byzantine Fault Tolerance, an improved version of Proof of Stakes in which holders of the coin have the power to validate blocks. In comparison, Ethereum currently runs a Proof of Work, as Bitcoin does, although it is scheduled to change to a Proof of Stake

system. There is also the previously discussed difference between them in terms of languages of programming, as Neo opted for several common languages in contrast with Ethereum. From these differences the most important one was the digital identity, as it would be a good introduction to these projects, as it allows for a more serious approach for the enterprises and a better reception from governments. Also, the digitalization of assets would be an interesting feature if the platform expands the concept of exchange of digital projects to real world assets and projects.

Lisk (17) is a recent project, founded in 2016 and forked of Crypti. In terms of blockchain technology differences, Lisk uses a Delegated Proof of Stake system and more importantly it has bet on side-chains as a solution for the implementation of decentralized applications. Ethereum focuses on securing the integrity of their blockchain through the introduction of Gas consumption to avoid malicious code and the security of their solidity language. Although this is fallible, as programmers are always imperfect. The most famous case is the DAO, an Initial Coin Offering that was hacked, allowing the hacker to take funds from the smart contract without diminishing his balance, due to a vulnerability in the code. This derived in the fork of Ethereum, where these transactions were cancelled, and Ethereum Classic, where they were kept. Lisk on the other hand opted for Sidechains for the decentralized applications build on top of it. This would mean that any security flaws in the programming of those would only affect them, something especially important considering that the decentralized applications in Lisk are programmed in JavaScript. In terms of public engaging Lisk is selling itself as an smoother platform to develop decentralized applications and more accessible thanks to their JavaScript system.

The last two options were not really considered as an option for the development because their technology is still in its infancy, even more when this project started, and don’t have yet a working product. Although it is certainly worth to mention, mainly because of the problem that they try to solve, scaling. Scalability issues are one of the biggest problems in blockchain technology, as it is a technology that relies on a shared ledger for all of the transactions, which leads to a poor performance in the number of transactions that they can handle. Currently the computational power of the Ethereum network is no more powerful than a single computer. Because of this limitations Cardano and EOS were created. In one phrase each, Cardano is a peer-to-peer review project that attempts to solve the scalability issues through multilayer computation. EOS on the other hand attempts to solve it combining more than one thread of computation for the whole network and continuously combine it. Although this innovations are for sure a great addition they haven’t been released yet and both projects don’t have a working product, so it is impossible to develop this project on them.

In conclusion, Blockchain as a whole is a technology still in its infancy and Ethereum is not an exception. But as can be seen with the four above commented projects, most of the improvements over Ethereum are not yet ready. Moreover, Ethereum is still in phase of great development, and nothing assures that these improvements are not going to be implemented in Ethereum, even before than its competitors.

## 4.3. Decentralized application interaction

The aim of this section is to present how decentralized applications (dApps) interact with the blockchain and with the users. Once this concept is clear there are several development options to support the front end of the application.

A decentralized application can work mostly as a normal web application. These last ones, receive data form central servers, instead, decentralized applications at some point connect with a blockchain, either to retrieve or write data. In the case of Ethereum the interaction with the blockchain not only can consists in this, but also it can ask for computation, the smart contracts. These contracts stored in the blockchain are immutable pieces of code, which functions are ready to be used by users.

From a user's perspective the design of the web page is presented in html code, rendered by the browser (CSS is also used for handling images and design elements, this will be omitted for clarity). This is what was available at the beginning of the internet, the web 1.0, but nowadays a web application is not only a statically designed web page. JavaScript introduces interaction with the web page (the web 2.0). This interaction handles passwords, emerging windows, interactive games, comments on social networks... Until this point centralized and decentralized webs work in the same way. By this time a normal application would connect with a central server, instead, a dApp connects with the blockchain thanks to the web3.js library. This library has the needed functions for these communications. Its name is not casual, as it pretends to be the next big innovation in how we understand the internet, creating the web 3.0.

But this library is not currently available in all browsers, and also if a user wants to use a decentralized application it needs to have a copy of the Ethereum's blockchain. For this missing step there are several options:

- Mist is the official dApp browser. Basically, is a full browser that already integrates the web3.js library and a local copy of the blockchain. Another similar option is Parity, which is another dApp browser.
- Metamask on the other hand is a Google Chrome extension. It adds to the Google Chrome browser the web3.js library and also connects the users with their own blockchain.
- There is also the option to develop a native application to host the interactions of the users. This would be a similar option to use the mist browser, as the user would need to also download the full blockchain.

When developing a platform is not necessary to choose between these options, but as a developer you can focus on supporting more one option. In this case, given the nature of this project, the Metamask solution seems the better suited. The only downside that it has is that the users of Metamask have to trust the blockchain of Metamask instead of having their own blockchain. Besides this, the functioning

of the platform can be perfectly integrated as a web page, instead of developing an application, and the user can start to run Metamask very fast compared with Mist (it takes time to download and install the application plus the time that takes to download the whole Blockchain).

## 4.4. Smart contracts

During the course of this report smart contracts had been mentioned extensively, this section aims to introduce their functioning through the explanation of a simple smart contract simulating the exchange of a project. This smart contract is developed in solidity and can be found entirely in the annexes of this report.

Solidity language is easy to understand to anyone with experience with any C language, Java or JavaScript. The basics of solidity can be found in the Ethereum´s website and also there are plenty of online forums. This overview on smart contracts will not aim to explain all of the options of solidity but instead show how a smart contract can govern the exchange of a project. The contract will allow to set a payment and a fee, will allow the client to select a creator to do the project once the payment has been deposited in the contract and allow to release the payment or call an arbitration.

```
1  pragma solidity ^0.4.18;
2  contract exchange{
3      address public client;
4      address public creator;
5      bool public cancelcontract=false;
6      bool public callarbitration=false;
7      uint public payment;
8      uint public fee;
9      uint public b=this.balance;
```

*Figure 5 Variables of the contract.*

The first line of the smart contract declares the version of solidity that will be compiled on. Then the contract is declared with a name, the contract, in this case “exchange” is more or less equivalent to an object in an object-oriented language, as the contract class is already built into the solidity language and has some predefined functionalities. But is also similar to a class, as you are defining a constructor, functions… etc.

As in a normal program, variables can be declared at the beginning of it. As it can be seen apart from the normal variable types (bool or uint) there is a address type, that represents an address of an Ethereum wallet. These addresses are used to keep and transfer funds, the contract itself has one address associated.

```
11     function exchange(uint newpayment, uint newfee) public{
12         client= msg.sender;
13         payment=newpayment;
14         fee=newfee;
15     }
```

*Figure 6 Constructor of the contract.*

The constructor of the contract is only called when the contract is created. In this case the constructor uses “msg.sender” , that is the address that is interacting with the contract in any given function, to store who is the client of the project and owner of the smart contract. When the contract is created two

variables are needed, the payment and the fee for the project. These three variables are created when the contract is first published and cannot be modified ever.

```
16      modifier ifclient(){
17          if(client!=msg.sender){
18              revert();
19          }
20          _;
21      }
22      modifier iffunds(){
23          if(this.balance==0){
24              revert();
25          }
26          _;
27      }
28      modifier ifcreator(){
29          if(creator!=msg.sender){
30              revert();
31          }
32          _;
33      }
34      modifier ifcancelcontract(){
35          if(!cancelcontract){
36              revert();
37          }
38          _;
39      }
40      modifier ifarbitration(){
41          if(!callarbitration){
42              revert();
43          }
44          _;
```

*Figure 7 Modifiers*

Solidity makes use of modifiers. A modifier can be placed in the definition of a function and it is called before the execution of the function. In this case the modifiers are used to check some logic conditions before executing a function. These conditions are written in reverse logic, and if they are met the function they are inserted in will not be executed (revert(); statement) or the function execution will continue ( _; statement). The modifiers included above check if the address that is calling a function is the address that was declared the client (ifclient) or if it is the creator (ifcreator), if there are funds stored in the smart contract (iffunds) or if some part of the process has been initiated checking some auxiliary Boolean values (ifcancelcontract and ifarbitration).

```
46      modifier ifpayment(){
47          if(msg.value != payment+fee){
48              revert();
49          }
50          _;
51      }
52      modifier iffee(){
53          if(msg.value !=fee){
54              revert();
55          }
56          _;
57      }
```

*Figure 8 Payment modifiers.*

Using msg.value it is also possible to check the value associated to a function, in this case a payable function, and check if the value is the predefined value. The two modifiers shown above check if a payment is equal to the payment plus the fee (this is the payment that should be done by the creator) or if a payment is equal to the fee (this is the payment that should be done by the creator of the project).

```
58      function client_funding() payable ifclient ifpayment public {
59
60      }
61      function creator_funding() payable ifcreator iffee public {
62      }
63      function getfunds() constant public returns(uint) {
64          return this.balance;
65      }
66      function setcreator(address newcreator) public ifclient iffunds {
67          creator= newcreator;
68      }
```

*Figure 9 Functions.*

After declaring the modifiers, the functions show how they are used. The first function is a payable function that uses ifclient and ifpayment to check if the funds that are being send to the smart contract by the client and is the expected quantity (the payment plus the fee). The second one checks the same, in this case for the client. The third one doesn´t make use of any modifiers, as it is just a public function that allows to see the funds stored in the smart contract.

The last function is used to store the address of the creator, designated by the client. This requires the creator to have already send the necessary funds to the smart contract.

```
69      function clientveredict(bool clientjudgment) public ifclient{
70          if(clientjudgment){
71              creator.transfer(this.balance);
72          }
73          else{
74              cancelcontract=true;
75          }
76      }
77      function creatorveredict(bool creatorjudgment) public ifcreator ifcancelcontract{
78          if(creatorjudgment){
79              client.transfer(this.balance);
80          }
81          else{
82              callarbitration=true;
83          }
84      }
85      function arbitration(bool arbitrationjudgment) public ifarbitration {
86          if(arbitrationjudgment){
87              client.transfer(this.balance);
88          }
89          else{
90              creator.transfer(this.balance);
91          }
92      }
93  }
```

*Figure 10 Process functions.*

The last functions show an example of a simplified agreement on the payment of the contract. In this case the client can say that the project has been satisfactory and the client will receive the funds of the project. Or he can try to cancel the contract if the expectations of the project are not met. But if he wants to retrieve the funds, the creator must agree on the decision, if that is not the case the arbitration will be activated. The arbitration puts the referee into the position of deciding if the project has met the expectations and he can choose who gets the money of the contract. In this case the arbitration function is a public function that anyone can activate, but in a real use case will be restricted to be used only by an authorised referee.

This system is very rudimentary but shows how it is easy to program a certain process of project development, how different permissions can be issued or restrain the conditions to protect the users. It also shows how the processes can be automated, for example selecting the referees by a program can be automated by another smart contract that sends safely selected address. This selection of referees will be discussed in depth in another headline.

Another interesting addition to this smart contract would be to implement a time restriction. Time restrictions in Ethereum are handled by checking the block number. As each block is being created each ten seconds the smart contract can easily include modifiers that allow actions, or cancel the contract if the agreed deliver time has elapsed.

Relating this section with the last one, the shown functions or variables are what constitutes the interaction with the smart contract. With the web3.js library a web can read variables or execute functions through their JavaScript program. Allowing the user to create the contract, or interact with it like he would interact with a normal web application. In that sense, working with decentralized

application is not necessarily different from our everyday interaction with the internet services. It can be even more easier to interact with if we take into account that it can handle payments without needing to introduce any data. When Metamask is activated the user must log into their wallet, having access to their identity and their stored cryptocurrencies. From that point, a simple interaction with a button can be a direct instant payment to any point in the world, stored in an immutable public data base and with a fraction of the cost of a bank transaction.

## 4.5. Token creation and ERC20

This section´s objective is to explain how to create a new token in the Ethereum Blockchain. A token built in Ethereum is powered by a smart contract. This smart contract defines the supply of the token (fixed or variable), the destination of the initial tokens and the functions that can be performed by the token.

```
contract MyToken {
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(
        uint256 initialSupply
        ) {
        balanceOf[msg.sender] = initialSupply;              // Give the creator all initial tokens
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) {
        require(balanceOf[msg.sender] >= _value);           // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        balanceOf[msg.sender] -= _value;                    // Subtract from the sender
        balanceOf[_to] += _value;                           // Add the same to the recipient
    }
}
```

*Figure 11 Simple token creation contract.*

This is the simplest token that can be created, is a fix supply token that can only perform a transfer function between users. An array of the different addresses is stored in the contract, the token share the address with the Ethereum address. With this contract we would have created an already integrated token as normal Ethereum's wallets will be able to handle token holdings and transactions. The functions in these tokens are treated equally as any smart contract function, as such the transaction fees are paid in Ether.

As it can be seen the token creation in Ethereum is a simple process well integrated with the Ethereum Network. A protocol to increment this integration has been developed a nowadays all of the different tokens follow the rules of this protocol. The name ERC20 stands for "Ethereum Request for Comments", these requests are the official way to propose an improvement in the Ethereum Network, the number 20 is just the ID of the concrete request. The ERC20 protocol implements the same functions for the tokens, allowing a better integration of these tokens in wallet or exchanges. An example of an ERC20 smart contract can be found in the annexes of this report.

The impact of the ERC20 shows the previously mentioned importance of belonging to a greater ecosystem. A place where this can be observed is in cryptocurrencies exchanges. Cryptocurrencies exchanges have to deal with several coin wallets, to allow their costumer to deposit and withdraw their coins. This fact has precipitated the development of two types of exchange, the ones that deal with the problems of money deposits (Coinbase or Bitstamp) and the ones that deal with the technical problems

of integrating a lot of different wallets but with no money deposits (Poloniex or Bittrex). But ERC20 tokens can be easily integrated in exchanges, allowing news like the recent announcement of ERC20 tokens by Coinbase (29). This is one of the main reasons why having a different token doesn't have to mean a big disadvantage, if it is a ERC20 token.

# 5. File hash storage

One of the main advantages of using blockchain technology is that it allows to use a permanent decentralized data base. At its core, the blockchain technology is a way to treat data in such a way that anyone can access and trust the data that is stored in it. This headline will explore the possibilities that this technology opens to project exchange platform and how can they be implemented.

This project can benefit from having a trusted data base as this would allow the users to prove their intellectual property on the projects that they develop.

But there are two things to consider when we want to store any data on a public blockchain:

- Any data stored on a public blockchain can be accessed by everyone. Then if any sensible data is stored on the blockchain it must be encrypted.
- The current state of blockchains based on Proof of Work, creates a scarcity of the available space on them. All the data written into the blockchain is secured and validated through the computing power of the network, this gives an intrinsic value to it so the data can't be infinite, thus making the data scarce and valuable. This generates a cost for any user who wants to write in a public blockchain. In Bitcoin for example each time a transaction is made you have to set a fee for each of the Kb that the miners are going to write into the next block. In Ethereum this cost is given in GAS.

These two restrictions limit the use of the blockchain use to store the files of the projects, which are both sensible information and are expensive to store, as their file size can be pretty high. The solution proposed in this project is to store only the hash of the project file. This solves both problems, because with the file hash the complete information of the project is encrypted and the storage would be only of a few bytes, depending of which hashing algorithm is used.

From the users' perspective this change form storing the file to storing the hash of the file is also beneficial. The only propose of storing the file as it was described before was to provide a trusted method of validating the authorship of a project. Having the hash of the file stored allows a quicker method of validating this, as it can be done easily just by comparing the hash of the file to the hash stored in the blockchain. Later on this report there will be a practical use case of this hash storage, were the referees can validate the information provided by the users through the information stored on the blockchain. In this case the process of validation is simpler as it can be done by simply looking to the hash of the file or it can even be automated with a simple program that is able to validate the file comparing it hash with the one that is on the blockchain.

## 5.1. Technical implementation

There are several methods of storing information in the Ethereum's blockchain. The main factor to choose one method will be the cost, this cost will be assumed by everyone of the users, so it is in the best interest of the platform reduce its cost of usage, until a point where the cost of this feature doesn't become a discouraging factor for using the platform.

```
pragma solidity ^0.4.18;
contract store
{
    string hash;
    function store(string h) public
    {
        hash=h;
    }
}
```

*Figure 12 Simple storage contract.*

The first method of storing information is through an interaction with the smart contract. This is the simplest way of storing information. Is done through a simple interaction with the smart contract. In the following images a couple of examples can be seen as how a hash can be stored in a smart contract, first through the constructor of the smart contract, simply setting it at the beginning, or later through a function. In the second case the function ensures that is only accessed by the creator of the smart contract (this can be modified to any particular address) and that is only possible to store the hash and not modify it, with a simple counter that enables only one interaction with the function.

```
pragma solidity ^0.4.18;
contract store
{
    string hash;
    address ContractCreator;
    uint counter;
    function store() public
    {
        ContractCreator=msg.sender;
        counter=1;
    }
    modifier ifContractCreator(){
        if(ContractCreator!=msg.sender){
            revert();
        }
        _;
    }
    function hashstorage(string h) ifContractCreator public
    {
        if(counter==1)
        {
            hash=h;
            counter--=1;
        }
    }
}
```

*Figure 13 One time storage contract.*

The advantage of this method is that enables some interesting interaction with the hash of the file, as it stored directly in the smart contract. A possible function is to require the client and the creator to store the hash of the final project, and check that is the same in the smart contract. But in order to evaluate a given technology it is important to take into consideration what are their main advantages, what we should be the use that we as engineers give them, and if there is some other technology that is more suitable. In this case the power of the smart contracts is to perform programs in a very secure way, backed by the Ethereum network. So, something simple as checking the hash of the file shouldn´t be a decisive factor to choose this method of storing the hash, as it can be done by the user himself, in the simplest example.

As mentioned in the literature review of this report, Ethereum solves the problem of the halting problem establishing a cost for the usage of their computational network. All of the instructions that smart contracts run have a cost in GAS, set in the whitepaper. GAS is an intermediate step into the conversion to ether, that helps to decouple the price of ether from the price of the use of the Ethereum network. So, the final price of the operation depends on how much you set as the Ether/GAS ratio. Currently the average price is 4 Gwei for each GAS or what is the same $4x10^{-9}$ ether. This average price depends exclusively on what transactions are the miners are accepting. This average price represents a price where the transactions are being accepted in less than a minute.

So, in the case of storing 256 bits in a Ethereum contract the price is 2000 Gas, this quantity is set on the Ethereum yellow paper. For obtaining the price in Dollars is necessary to multiply this quantity by the Ether/GAS ratio and the Dollar/Ether ratio. Nowadays these ratios are $4x10^{-9}$ ether for each GAS and 1000$ per ether. So, the final price of storing the hash in the smart contract is 8 cents of Dollar or converting it into Pounds, 0.05 £.

$$20000\,GAS \times 4 \times 10^{-9}\,\frac{Ether}{GAS} \times 1000\,\frac{\$}{Ether} = 0.08\$$$

This price is sufficiently low to be considered this method as a feasible implementation into the platform. Also, it has to be considered that this is the simplest method of implementing the hash storage. Although another method is going to be considered in this report.

This method consists in storing the hash of the file into the data of a transaction. The data of the transaction lets the user store information as a complement to any transaction of the platform. This method has two mayor drawbacks in comparison with the previous one:

- Unlike the smart contract storage, this data can't be accessed from the smart contract, but can be displayed from the blockchain data. As it was discussed previously this drawback alone shouldn´t discourage the use of this method. The problem is that storing the hash into a transaction will cause that to retrieve the data a user must know in what transaction is it. This

would mean that the users involved in the project would no longer be anonymous. This can be a major drawback if we consider the necessity for anonymity in the arbitration process for example.

- Another drawback is the fact that only lets users store the file of a hash when doing a transaction. This means that the process flow of the platform will be restricted, to require the hash of a file, only when a transaction is made. For example, to require the hash of the project's file to the creator only when he makes the transaction for his fee. While this is not an excluding characteristic for this method it is certainly something to consider against it.

Regarding the cost, the cost of each non-zero byte written in a transaction is 68 GAS. So for 256 bits, or 32 bytes the final cost would be:

$$32\ bytes\ \times 68\ \frac{GAS}{byte}\ \times 4 \times\ 10^{-9}\ \frac{Ether}{GAS}\ \times 1000\ \frac{\$}{Ether} = 0.0087\ \$$$

The price is roughly one tenth of the one with the previous method. It is worth mentioning that as the approach would be to implement this date into a necessary transaction of the process the cost of the transaction is not taken into account as a cost of storage, because this cost had to be assumed by the user anyway. This cost would be of 2100000 GAS.

Taking into consideration of all the above arguments the best technical solution for the hash storage seems to be storage the data directly in the smart contract. The main reason is that it has no technical drawbacks compared with the second method, mainly it allows to keep anonymity. And the cost difference although is very big comparing the two of them it has to be considered that both are still very cheap methods, that don't interfere with a cheap functioning of the platform.

# 6. File Sharing

One of the main features of this platform will be to provide a secure way to handle all of the information exchange in the development of projects. In this headline it will be discussed how to provide this feature to the user. The method chosen to be implemented into the platform is a peer to peer (P2P) file sharing tool. The first topic that should be addressed are the reasons behind this choice.

While the standard approach would be to implement a server where the users are able to upload and share with other users the content of their projects this creates two contradictions with the aim of a decentralize platform. First of all, it creates a centralize cost for the company, the servers. This cost is one of the many costs that justifies a charge for the usage of a platform. For example, Airbnb, although is called sharing economy, at the end is yet a centralized service as a company runs the platform and charges a percentage of the interchanged value as a payment for the service that they provide. In this platform the approach should be different avoiding as much as possible a centralized cost for the platform.

The second aspect is also about decentralization, as a central server would mean that it can be security breach, and the users' information can be hacked. Although the general public is yet starting to be concerned with online privacy, companies are already aware of how important is to ensure their data is private. In that sense the current approach in most of the cases is to rely on a trusted service to take care of the security. One of the most important aspects of Blockchain technology in that sense is that it encourages the users to take care of their own security instead of trusting others. So, the P2P file sharing solutions fits this philosophy as it is not dependant on any central authority and ensures the security of the users' data.

The main drawback of this method of file sharing is that it requires both users to have connected at the same time their computers to exchange any files. Although the system for the file sharing can be implemented to allow to initiate the file exchange without being in the computer. To overcome this problem there were two alternate solutions, enabling a decentralize file sharing tool but not a pure P2P file exchange. The two possible alternatives were a solution based on decentralized cloud storage, already covered in the literature review, and to implement a system with connection to the Inter Planetary File System (IPFS). The option to use decentralized cloud storage was dismissed because although it is pretty cheap compared with the traditional servers and still count as a decentralized solution, it is still a cost and a solution that adds more complications to the platform just to avoid a small drawback. On the other hand, IPFS, works different from a decentralized cloud storage system. IPFS is an internet protocol where instead of accessing a web page that is located on a server, it uses a peer to peer system of distribution. Each time a user access an url the content is stored in their computer and serves as a peer to the other users. The problem with IPFS is that is a very powerful tool, that could be adapted to this platform, but it is still in its infancy. So, it happens the same that happened before with

the decentralized cloud storage, it would be a very big challenge to implement IPFS to allow users to exchange files even when they are offline, just to avoid this drawback.

As many times in engineering, the best decision is to implement a solution of compromise. Although the main file sharing tool to be implemented in the platform is a purely P2P file sharing application, there will be another option in case it is particularly difficult for two users to share files through the P2P file sharing application. This can be done through centralized or decentralized servers, as the users who decide to use it are renouncing to use a purely decentralized platform.

Once the choice is clear another point that needs to be highlighted is why it is important to implement in the platform a native file sharing application. This is important because many of the file sharing applications named in the literature review have available APIs that could be used to use them for the platform. There are three main reasons behind this decision:

- One of the main focuses of the design of the platform is its independence from third parties. In this case this philosophy can be applied choosing a native application over using the APIs of an available one. The main benefit from this it that it makes the platform less dependent form other platforms, something that has to be considered when designing a platform in which the people will depend for a project to be completed or economically.
- For security reasons is also a good practice to only outsource work that it cannot be implemented in the platform.
- The file sharing system will be interacting with the rest of the platform, storing the hashes of the files in the system, confirming the completion of a project to a smart contract… So, in that sense it is better for integration of the whole system to develop a native application in the platform.

# 7. Arbitration

An arbitration system will be designed with the goal of preventing any misuse of the platform. As presented before, through smart contracts the client will have the option of cancelling the project. If this happens both parties can be satisfied, the creator of the project can accept that his work has not met the specifications of the project. The arbitration system will be implemented to prevent an unfair use of this feature, giving the creator a chance to call for arbitration if the contract is cancelled.

The arbitration system of the network will be designed with freelance referees. This is mandatory if the platform aims to be as decentralized as possible. To enable such an arbitration system rewards must be granted for the referees. The dilemma is where are the coins rewarded coming from. The option of a system where the rewards are granted by the platform is excluded, as this would mean that the arbitration system is a centralized service provided by the platform, consequently creating a mandatory charge for using the platform. To avoid this there are two options:

The coins rewarded can be created each time they are needed. This is possible due to the nature of the tokens, that can be minted through an interaction with the smart contract that regulates them. The biggest problem would be designing the system of minting coins well enough, in a way that they can be only minted when a successful arbitration has taken place, avoiding scams in which there are fake projects, hosted by a user with several accounts to create coins. Apart from this technical challenge, the introduction of new minted tokens would create inflation in the system, devaluating the coin in the long run.

The second option is to leave the responsibility of covering the cost of the arbitration to the users of the platform. The arbitration cost, referred as fee from now on, will be assumed by the user who is determined to be doing an unfair use of the platform. For example, the client if he has cancelled a contract although the project was successfully completed, or the creator if he claims the reward although the requirements of the contract haven’t been met.

This option has the following advantages:

- It creates an economical drawback to a user that wants to act maliciously. In cases where this use is obvious the arbitration will prevent them completely.
- It leaves freedom to the users to set the fee. Different tasks have different arbitration complexities, and the fee should be set accordingly to this. It is in the interest of a client to set a fair fee, so if it is needed, good referees will sign up for arbitrating the dispute. Also, if the client intends to use the platform fairly he will receive the fee back. On the side of the creator, the fee will be a deciding factor to take or not to take the contract, in that sense is the same as the payment set by the client, if the fee is too low the client should avoid the contract, as it may indicate an intended misuse of the platform by the client.

- This system of fees also creates a store of value in the token. For example, if a client receives back the fee after a completed project in Projectcoins it is likely that they store them. This represents two big advantages, first of all it encourages the future use of the platform, once it is used. Secondly, it distributes the token among the users, this means that more tokens are in the hands of user rather than in the hands of speculators, lowering the volatility of the token while reinforcing its spread as a store of value.

For these reasons, the arbitration system will be based on the users' fees. Once this is established, the possible outcomes of an arbitration will be presented.

If there are no arbitrations required there are two possible outcomes, the project is completed or the project is uncompleted. In both cases the fees will come back to the users, and the payment will go to the creator only if the client considers that the project is completed and according to the requirements.

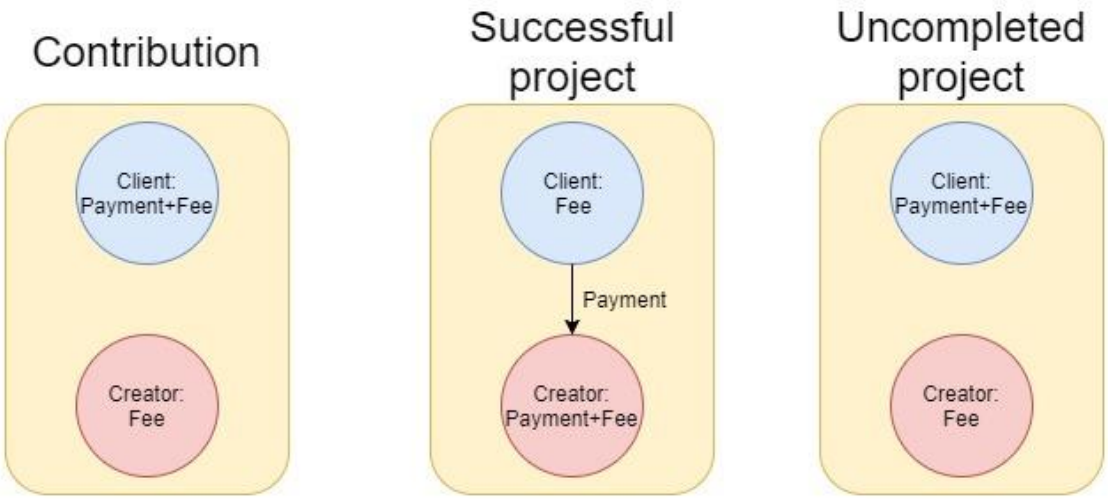



*Figure 14 Project development diagram.*

This is the initial approach for the project payments, but this can be changed to give more options for the users. For example, for avoiding scams in the smart contract is easy to implement a restriction that requires the client to send the payment before allowing the creator to send the fees. This can be done through modifiers inserted into the functions. A modification of the previously described functioning would be the requirement of the creator's fee, this can be done to ensure that the creator delivers a working project in time, and have the creator's fee as a commitment for it.

Moving into the actual cases where the arbitration is called, it has to be noted that the arbitration system will be always initially called by the creator because the client will have the option to not pay the project if they consider them unfinished. In this case the arbitration system is very simple, the referee will have access to the specifications of the project, the file that has been exchanged as the final project and both users can present allegations to the referee if they consider. When the contract is created the time for the

arbitration must be chosen. The smart contract will require the referee to dictate a verdict before this time elapses, once the verdict is dictated the smart contract will give the option to the user who has lost the arbitration to agree on the verdict, from here there are two options:

If the user who loses the arbitration agrees on the verdict the funds of the smart contract will be released. The result depends on which user have lost the arbitration. Not only because of the payment, but also because the reward of the referee will come from the user who lost the arbitration.

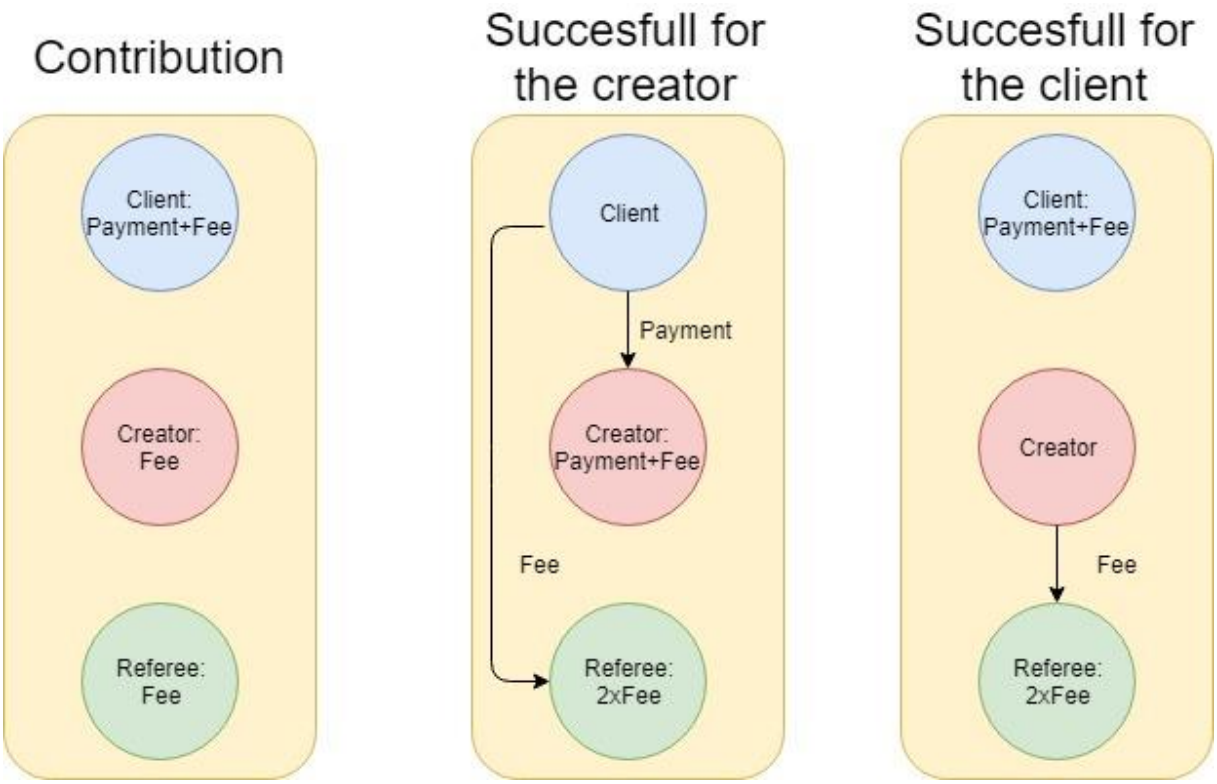



*Figure 15 Arbitration payment diagram.*

On the order hand if a user consider that the arbitration has been judged wrong he will have the option to ask for a second arbitration. This will be implemented to ensure that the arbitration of the project doesn't depend entirely on a single person because this will mean that the system is more vulnerable to any flaws in the security of the selection process, explained in the following section.

This second arbitration is the reason to require the referee to send the fee to the smart contract. The fee of the first referee will be only used in this second arbitration, to pay to the new referees if the second arbitration dictates that the first arbitration was wrong. This fee will serve not only as a payment to the new referees but also as an incentive to the first referees to dictate good verdicts and to judge only projects that they can fully understand, not only to receive money but for not losing them.

The second arbitration will require again a selection of new referees. The arbitration system proposed in this report will consist of two judges, although it was considered to implement the second arbitration with tree referees. The main difference between two and three referees is if the first referee verdict is

taken into account or not. The reason for this is that the system has to resolve potential verdict ties. If there are two referees in the second arbitrage the total number of referees is three and the first referee verdict can be taking into account. On the other hand, with three referees in the second arbitrage the total number of referees is four, thus making ties possible if the first referee verdict is taken into account.

Once this is explained there are two main reasons why the two extra referees' option has been chosen. First of all, with three extra referees a verdict that has two referees opposing the first referee and one baking him would have a contradiction, because besides the total compute of the four verdicts is a tie the final verdict will be the opposed to the first judge. The second reason is that the first judge shouldn't be invalidated because one of the users disagree with it. Although the users have the option to request more opinions, is not fair that just because the verdict is not beneficial for one user he has the option to disqualify that referee and effectively restart the arbitration process with other three judges.

The payment for the second arbitrage has to take into consideration the two extra referees that must be paid. To fund this payment the users that has lost the first arbitrage, and asks for the second arbitrage, will have to pay two referees´ fees. In the case that he lost the second arbitration also this money will go to the two referees. On the other hand, if he wins, and the first arbitrage is determined wrong, the first fee of the other user and the fee of the first referee will pay for two new referees. This way the user is wright doesn´t ever pay anything and the first referee has an economic drawback.

## 7.1. Selecting the referees

An arbitration system with freelance referees has to be able to select between the candidates that present themselves to arbitrate the project. Selecting the referees is critical to have a successful arbitration. The main goal of this system is to avoid that a user scams another user through the platform, by becoming the referee of his own project. To avoid this the goal is to implement two factors of security, reputation and randomness.

First of all, randomness introduces a factor that will discourage any misuse of the platform, because the scammer will never be sure that he will get elected referee. The problem is that if there is only a random selection between the possible candidates, malicious users can create several accounts to cheat the system and have more chances to foul the system, even if presenting to arbitrate a project means upfronting the payment of a fee. To solve this a reputation system has to be introduced. This reputation system is easy to design. As a referee reputation has to mean just successful arbitration as it has no meaning to implement a system where the users can vote how satisfactory the arbitration was. So, every account in the network will have an associated reputation and when the referee selection process begins the system will select randomly a referee between the users that want to arbitrate that specific project and have more reputation. For example, selecting a random user between the five that have more reputation from all the users that have shown interest in the arbitration.

From a technical standpoint of view there are to approaches to implement the system. The first one is to implement the system in a centralized way, having both the reputation administration and the referee selection in hands of the developers of the project. The advantages of this system is the traditional way that this things are handled nowadays and it should be easier to adopt any necessary changes if any flaws in the system are detected.

The second way of implementing a reputation system is to create a token to represent it. The idea is to create a token that has no real value besides to show the reputation of a user in the platform. To implement this token, it has to be taken into consideration two special things that differ from a normal token. The reputation token won't have a transfer function that enables the transfer of the token between users, this will avoid trading with them and will ensure that a user have earnt any reputation tokens that his account have. The minting process of this reputation token has to be only available to be called by a smart contract when an arbitration is successful, or to the developers of the platform to emend any possible flaws of the system. The advantage of this implementation is that is more decentralized and the system will work more independently from its developers. Also, if the system is well designed the token approach will give more security to the reputation tokens.

In the future development of the platform the initial approach will be the traditional one, until the project network is up and running flawlessly, at that point the system will be changed to the decentralised approach.

## 7.2. Referees access

Once the referee is selected, he has to be able to perform the arbitration process. For this he must have access to the smart contract verdict system and the specifications and final file of the project.

Having access to the verdict system means that the account of the referee has to be able to send a verdict to the smart contract. This will restrict any other person to emit a verdict. Looking at the smart contract this is an easy task. Restrictions are made possible in solidity, via modifiers, that had been introduced before in this report.

Since the platform is decentralize the file transfer to give the referee access to the specifications and final project has to be performed by the users. The proposed system is the same P2P file sharing service provided for the rest of the platform. The user in charge of this file transfer will be the one that calls for an arbitration, either the first or the second. The two files, with the specifications and the final results, can be verified by the referees in the smart contract, as their file hashes had been stored in the smart contract, this is one of the main reasons to store them as it allows to avoid scams with fake files. The referees must ensure that the files are the correct ones because is in their best interest to avoid scams, that would lead them to new arbitration and a loss of money for them.

## 7.3. Further considerations

To ensure a good functioning of the arbitration system the ideal is that the addresses of the clients and creators remain anonymous. The reason for this is that it disables possible communications between them and thus possible bribes. In order to do this, smart contracts allow to make private the accounts of their users. Accordingly, this will be implemented in the smart contracts of the platform.

# 8. Economic analysis

One of the main advantages of the development of a platform that uses blockchain technology is the facility to approach its funding. This is due to two main reason, both of them related to Bitcoin's nature. First of all, Bitcoin, and all the alt coins that have followed it, are an immutable digital asset, stored as information on a distributed ledger. This allows them to have value, as they can't be duplicated. Every platform that creates their own token can be funded through a public crowd sale of them, in an ICO (Initial Coin Offering). An ICO works as a funding method, opening the possibility to investors to invest in the development of a platform or project in the hopes that the total wealth stored in the token will rise in time and thus their investment return.

The second reason is the availability of investment capital in the crypto space. The impressive growth of almost all cryptocurrencies has created a dynamic investment environment, where the investors have seen great returns and are willing to be the early investors in new projects.

Besides this availability of funding opportunities, the design of an ICO shouldn't be done thinking in the availability of funding money but instead in the real necessities of the platform. Trying to adjust the ICO to a realistic economic plan for the development of the platform and design the whole process keeping the token valuable for the investors should be the priorities in the design of an ICO.

Another important characteristic of the ICOs is that not all the value collected by the developers is coming directly from the investors. There is also value in the form of tokens held by the developers and not sold in the ICO. This characteristic is often used by the developers to reward themselves, if the token succeeds, with funds that are not supposed to be invested in the development of the platform. But in this case this characteristic is going to be also used as a promotion method, keeping tokens away from the ICO and using them to promote the platform among companies that wish to offer projects in it.

This headline's main goal is to design an ICO accordingly to the previously mentioned objectives. To do this an economic analysis will be carried out in several levels:

- Determine a reasonable time frame to target as a period of development, during which the platform will be sustained from the funds raised in the ICO.
- Do an estimation of the costs of developing a platform like this would incur. This is perhaps the most important estimation as it will determine what are the goals for the ICO.
- Brief study of the actual ICOs to know if the expectations are realistic.
- Possible market cap for the platform, considering similar platforms that are already working. This is very important to determine the total funding of the ICO, so there is still room to grow and offer a good investment option.

- An analysis of the long-term sustainability after the initial funding money is depleted and the development of the platform has to be dependent on the user's contributions.
- Design of a model of promotion among companies that relies on held tokens that adjust both to the growing prevision of the platform and that preserves the value of the token.

## 8.1. Technical overview

Before starting with the proper economic analysis as it is based mainly in the ICO of the project, it is important to high light the actual functioning of an ICO. An ICO is the process where the Tokens that are created, as seen in the coin creation headline, are distributed among the investors. Explaining this distribution is the main goal of this section.

The most common way of doing an ICO of an ERC20 token is through a smart contract. As it can be imagined it is the simplest way to perform it, as it permits the exchange between ether and the new token while being able to implement restrictions to ensure the well-functioning of the ICO. In the annexes an example of an ICO smart contract can be found. Below a broad explanation of its characteristics can be found.

The basic function of an ICO smart contract is to receive ether from anyone willing to invest and to give back the new token. This can be done with only a function. In this case the below function has no name, as it is the default function that executes whenever any address sends funds to the smart contract.

```
function () payable {
    require(!crowdsaleClosed);
    uint amount = msg.value;
    balanceOf[msg.sender] += amount;
    amountRaised += amount;
    tokenReward.transfer(msg.sender, amount / price);
    FundTransfer(msg.sender, amount, true);
}
```

*Figure 16 ICO main function (30).*

In this function is comprised the whole operation of the smart contract. The operation of this function is to receive a payment and store its value, add this amount to keep track of how much has been raised in total and reward the investor with the tokens.

The line “require(!crowdsaleClosed);” checks this variable to ensure that the crowd sale continues. In this case there are two reasons for the crowd sale to be over, that are encoded in the rest of the smart contract.

The crowd sale has reached his goal. In this case all the tokens had been sold and there is no more possibility to invest in the ICO. On the other hand, if the target has not been reached and a time limit has elapsed the crowd sale is considered failed and the funds are returned to the investors.

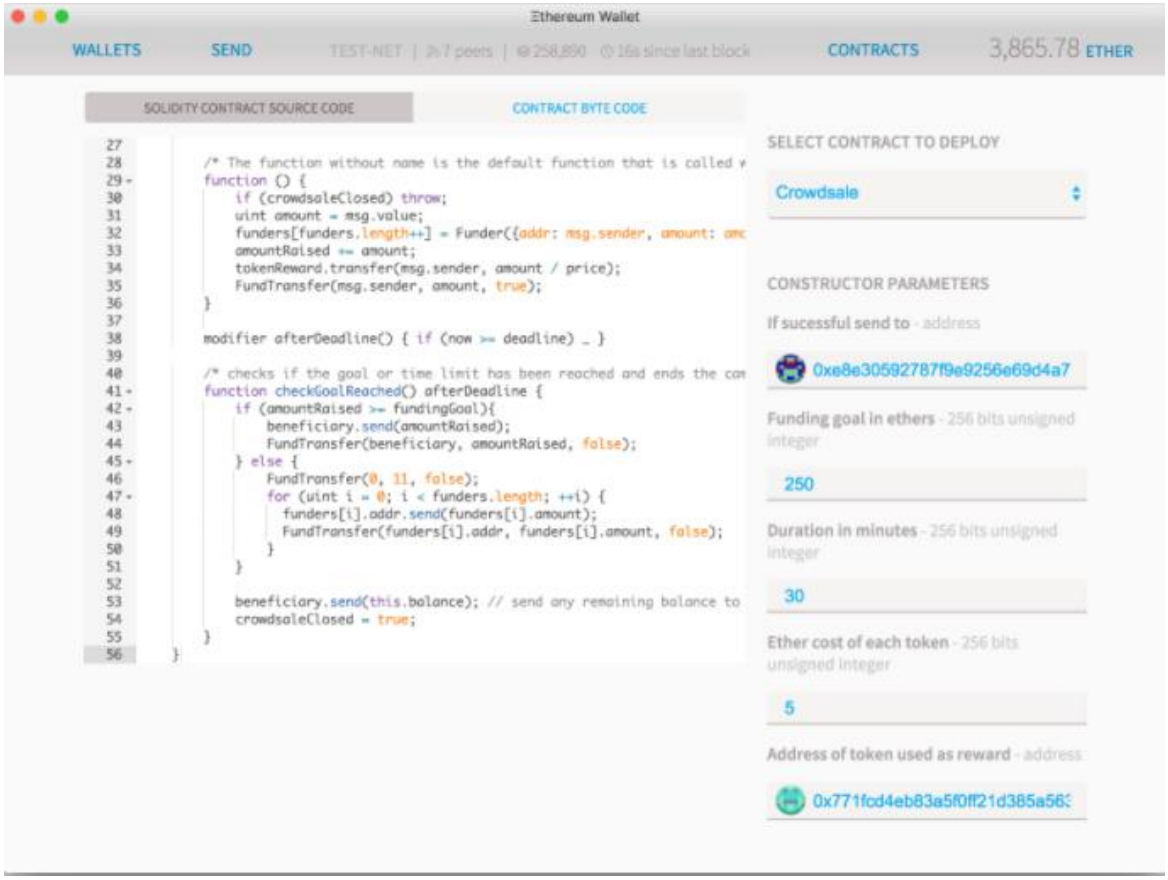



*Figure 17 ICO contract interaction (30).*

In reality, most of the ICOs don´t function with single target. Instead there is a first target, that has to be fulfilled in order to consider the ICO successful and retain the funds. Then there is a hard cap, over which no more payments are accepted. This model imposes a difference with the smart contract of example, as there are no longer two fix possible outcomes of the ICO, the ICO can be fulfilled anywhere in between the first target and the hard cap. To handle the usual procedure is to entitle the smart contract of the ICO as the owner of the token creation smart contract, allowing the crowd sale to mint more tokens as they are being sold, this avoids creating more tokens than those that are necessary. Allowing this type of ICO requires as mentioned before to create a mintable token in the first place, change the ownership of the original smart contract to the new ICO smart contract and substitute the transfer function for this mint function “tokenReward.mintToken(msg.sender, amount / price)”.

As it can be seen an ICO performed with Ethereum has as much rules and security as the developers want to offer, or as much as the investors push forward to have. So, although an ICO is not a regulated investment, honest developers can polish a secure ICO that protects the investors taking advantage of the possibilities of the Ethereum network. Nowadays ICOs are usually performed having in mind this and a common practise is to lock the funds of an ICO until certain time has elapsed or some conditions are met.

The next step in this direction are the DAICOs a model recently proposed by Vitalik Buterin (31). This model is based on the idea that investors should have some control over the project. This control is basically translated into two powers, controlling how much ether is accessible to the developers and having the ability to shut down the project and give back the remaining ether to the investors. These two powers are granted through a voting system that grants votes to the token holders. This new system is yet to be tested in real life ICOs, and some flaws can be easily seen, for example: how to prevent that some party (developers or large exchanges) with sufficient funds control this mechanism for its own benefit? What if the project is delivering the agreed development but the market price for the token drops below the ICO price? Or on the other hand, it is useful the self-destruction of the ICO when the market price of the token is higher than the ICO price? Nevertheless, DAICOs are an interesting concept that can become the standard in the very near future, so it is worth to mention.

Once the technical part has been overviewed the important aspects to the economic analysis are that defining an ICO has two very important factors to set, the two targets. The rest of the economic analysis will take this into account and will not only try to define a realistic first target, the minimum amount of money that has to be raised in order to kickstart the platform, but also a realistic hard cap, that prevents the project from being overfunded.

Another thing to consider, not mentioned before, is the fluctuations in price of the funds received in the ICO. Only in 2017 Ethereum's price has risen by over 5700 %. This fact as important as it is cannot be taken into account accurately in the estimations of this economic analysis. However, the estimations are going to be treated with a security margin, to overcome potential losses due to the fluctuations of Ethereum's price.

## 8.2. Time frame

The crypto space is a highly changing technological and economic environment. As it was reviewed in previous headlines of this report, even well stablished players of the market such as Bitcoin or Ethereum are facing not only direct competition, but greatly different technological proposals. Combined with the high volatility of the market, this makes a very uncertain scenario in which to try to propose economic estimations. However, as it was mentioned several times in this report, the vision of this projects sees itself as a project build over the existing blockchain technology, making a use case of it. Being consequent with this idea implies that this project is a bet on the success of the blockchain technology and the uncertainties surrounding it have to be accepted.

Different projects have different road maps, depending on the ambitions, technical challenges and business model. Is out of the scope of this report to make an exhaust comparison of the different time frames of other projects, but as an example the roadmaps of two projects mentioned early on this report are going to be overviewed.

Siacoin has an ambitious roadmap, since its crowd sale in 2014 there were two years until the beta came out, another two years of technical improvement for improving as an option for cold storage and two final years of expansion among companies finishing 2020 as a serious competitor for Amazon S3 as a warm storage option. Although the objective of Sia coin is very concrete, it has many technical difficulties to overcome. Also, their ambition is to take over the growing market of cloud storage, competing against the tech giants, and Sia coin has a decentralized business model where the main beneficiaries are the users that rent space on the network, not the development team. Having these three considerations makes sense that their road map has such a long-time frame.

First Blood on the other side has a development cycle of only 18 months. Compared with Sia Coin, the three factors analysed for it are completely different for First Blood. There are no big difficulties to overcome in the case of First Blood, as an ERC20 token build over Ethereum the project is not facing any big technical barriers. The ambition of the project is smaller, as gambling in eSports is not yet a very big market with important predominant players. Finally, their business model includes charging a 5% to the bets of the players and also revenue from advertising in their tournaments. These three aspects combined make First Blood legitimately attempt to be up and running in a very short span of time.

In the case of this project it is somewhat in the middle of the two previous mentioned projects. With First blood it shares similar technical challenges and with Sia Coin it shares a decentralized business model. In the ambition aspect, this project can be considered to be in the middle, as it doesn’t face any major players like Siacoin, but the mission of this project is much more disrupting than what First Blood aims to provide. For these reasons a time frame of 3 years seems to be realistic enough to develop the platform and have margin to adapted well enough to be successful and reach a critical mass.

## 8.3. Cost estimations

For the reasons given in the previous paragraph, this section´s main goal is to set a minimum viable inversion from the ICO that allows the project to be developed for a time frame of 3 years. These costs are going to be projected for the company being based in Spain.

First and most important of all the major cost that has to be estimated is the spending in human resources. The minimum viable team for developing this project is going to be considered to be formed by 3 people. Two of them of a technical profile, in order to be able to develop and maintain the platform. It would be risky to aim to develop a platform like this one with all the responsibilities falling on a person, different points of views are very important in any engineering development. The third person should be of a marketing/economical profile, able to abstract from the technical problems and have a global vision of the project development. This is especially important in this project, that is focused to offer a solution to companies, and this should be a priority in the three years of development.

Assuming for them the median cost of a worker in Spain the cost of each one of these workers would be 30,000€ each year. That accounts for a total of 270,000€.

Having such an important cost invested in human resources is not optional as it was mentioned before. But as this is an estimation of minimum viability other expenses can be reduced.

First of all, the cost in materials for the development of the project can be minimal in comparison with the human resources expenses. As a minimum viable estimation, the project can be assumed to resort to telecommuting in order to reduce the expenses from renting an office.

There another two important costs that can be reduced thanks to the nature of the project. The first is all the cost of little services that the developers may need. For example, the creation of a logo for the network can be outsourced via the own project network, using the created tokens. This would be a stimulation to the network, while keeping possible to continue with the development. Another cost that can be reduced is the publicity that can be trusted to the token distribution method that is going to be explained at the end of this headline. Using tokens to promote the platform should be enough to reduce the publicity cost to the minimum.

With all of this consideration taken into account the expenses of the project can be estimated in 270,000€ plus 100,000€ for covering the other expenses. In total 370,000€, or what is the same 455,000$.

## 8.4. Actual ICOs

This section is not an exhaustive study of the actual ICOs but instead just a look into how much money can be expected to be raised via this method. The first interesting data to look is the whole size of the whole ICO funds raised. Money raised via ICO has recently surpassed the early stage VC funding for start-ups (25) and it is continuing to grow as it can be seen in the following graphs.

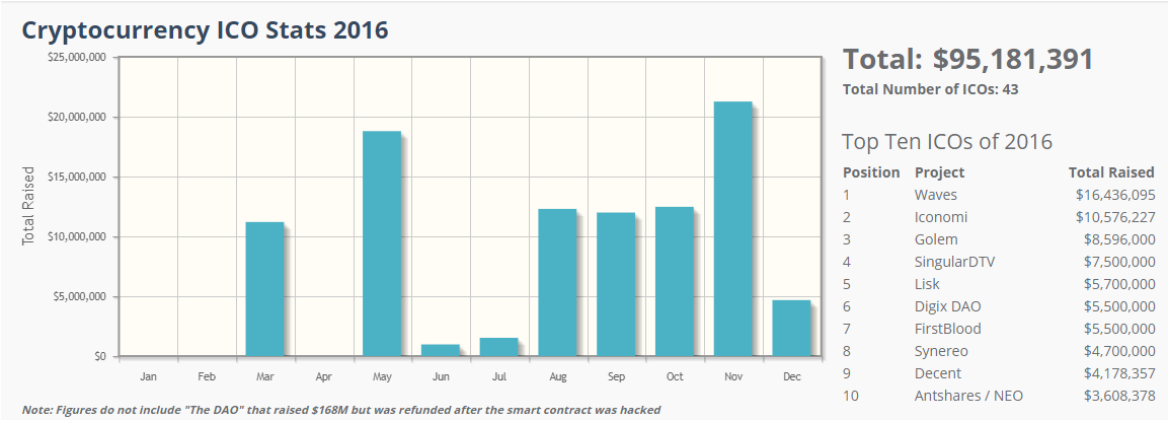



*Figure 18 ICO funding in 2016 (32).*

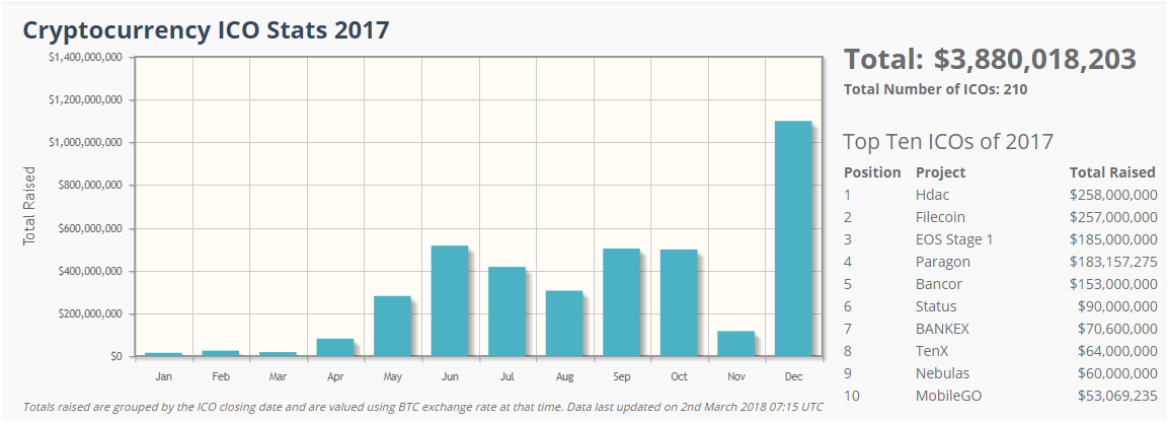



*Figure 19 ICO funding in 2017 (31).*

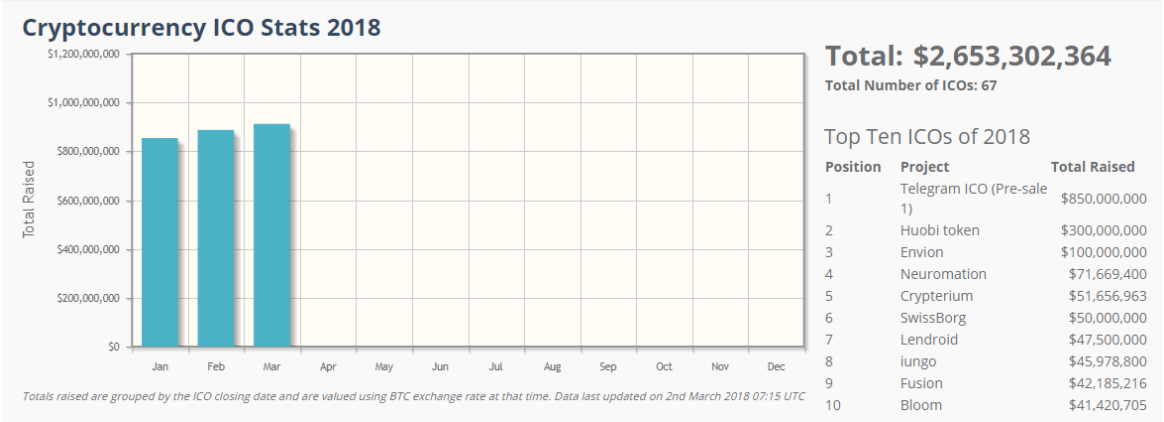



*Figure 20 ICO funding in 2018 (31).*

The first thing that can be noticed is the continuous growth experienced by the ICOs, fuelled thanks to the impressive growth of the crypto currencies, but also thanks to big companies running ICOs, such as Telegram or Huobi. The ICO of Telegram alone is predicted to raise up to 2 billion dollars, so 2018 is expected to continue with this growth.

But in order to analyse the viability of the objectives of an ICO for this project, is more important to analyse how the raised money is distributed among the different ICOs. From the 310 ICOs analysed in CoinSchedule (32) there are 9 with less than 100,000$ raised, 32 that raised between 100,00$ and 1,000,000$, 102 that raised between 1,000,000$ and 10,000,000$, and the rest, 166, has raised over 10,000,000$.

From this data it can be seen the potential funding option that represents an ICO. It has to be noted that the 9 ICO that raised less than 100,000$ had taken place before the spring of 2017, where there was a major price increase in Ethereum and as it can be seen in the graph of 2017, it impacted directly to the total investment in ICOs.

In conclusion, this data shows that only 13.2% of the ICOs raised under 1,000,000$ and if we take the most recent data, every ICO is raising at least 100,000$. These data combined with the previous section´s estimations are very promising towards the viability of the project. This will be discussed in the last section of this headline.

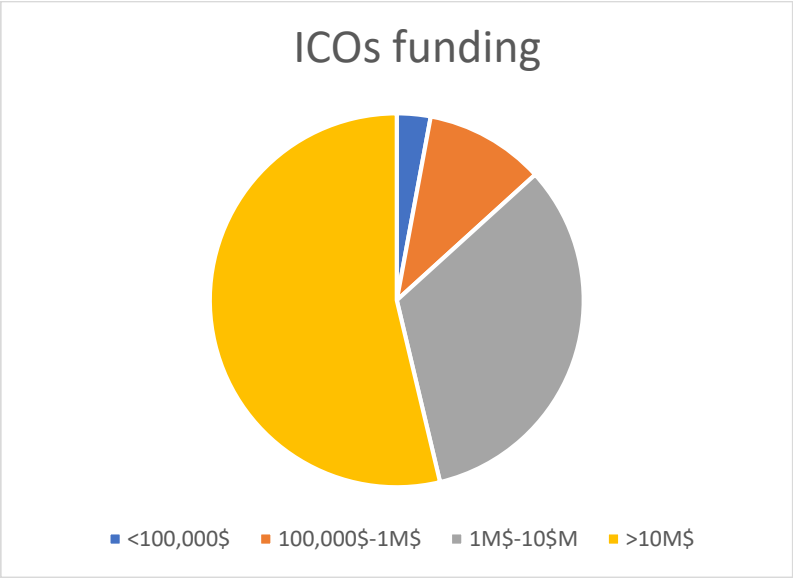



*Figure 21 ICOs funding.*

## 8.5. Market capitalization estimation

The total market capitalization is a very important measure in every blockchain project. It is the result of multiplying the market price of the token times the token total supply, it represents in essence how much wealth is stored in a platform. In the particular case of this platform it represents how much wealth is being exchanged in the platform. So, this estimation will have two main uses, have a base to calculate a long-term sustainability for the project based on a charge to the users and plan the ICO responsibly to have a real growth potential. It is because of these uses that the following estimation is a mid-term estimation.

To obtain a reasonable goal for the project the main source of information will be the existing online resources available to do online jobs. Although there are differences with these online marketplaces, mainly being that this project aims to target companies that are interested in a reliable system of project exchange, thanks to the blockchain tools that are described in the previous headlines. Thus, being the target of this project to have less transactions but with a bigger economic reward. Although these services are consequently very different from this one, this project can be seen as an evolution of them, thanks to a new technology.

The first online market place to be analysed is the Amazon Mechanical Turk. This project was a tool created by Amazon to be used by themselves that later was opened to the public. It is focused on small tasks that computers are not able to do, for doing them the reward is usually less than a dollar. Although it targets a different public it is interesting to be analysed, because is powered by one of the biggest tech companies. This online estimation (33) states “These numbers generate a yearly transaction volume for Mechanical Turk between $10M and $150M.” It has to be noted that this estimation is from 2012, but it is a good starting point, because as explained before the objective of this section is to provide a mid-term estimation, not full potential one.

The second company to be analysed is fiverr, is interesting because unlike other online marketplaces it has a focus on projects and not only small tasks like online questionnaires. The name of the company comes from the fact that the minimum reward for each task is five dollars. They generate 1 million tasks each month (34). There are not official statistics about their mean prize per task, but with a minimum task prize of 5 $ and a focus on being a reference market place for bigger tasks it can be assumed that the mean prize could be in the 10$-20$. This gives a yearly volume of $120M-$240M.

Having a look to the general situation, there are great prospections for the gig economy in terms of general adoption. This would be beneficial to this project that has a focus on being a reliable platform for project commissioning, creating a new market niche for the gig economy. Taking this into account and the two above estimations, of Amazon Mechanical Turk and fiverr, the annual market volume for this platform in the mid-term can be estimated around the 10M$-100M$.

## 8.6. Long-term sustainability

As a decentralize platform, the economic aim of the project is not to maximize the profits, but instead to achieve a state of self-sustainability where the platform is able to be guarded by a team of developers. The necessary income for this long-term sustainability will come from the users of the platform. Although it is impossible to assure at this stage if that long-term sustainability is achievable it would be irresponsible to launch a project involving public investors in a project that hasn't been critically evaluated form an economic point of view.

The two most important data to analyse the sustainability of the project are its projected volume and the revenue from the existing online job services. The main source of revenue of the online job services is a direct charge on the users. This charge varies from 10% up to 20%. This is a huge charge to the users that can be even larger. In the case of fiverr on top of the 20% charge there is a 0.5$ charge to the users that post a job proposition, resulting in a cost for the user of 5.5$ and a reward for the other user of just 4$.

Taking the conservative estimation of a 10M$ volume and charging a 10% charge to the users would mean an annual revenue of 1M$, two times the estimated required revenue for a 3 year long period development. In conclusion, a standard charge on the users would enable the sustainability of the project with a conservative projection of the market volume. This would allow the sustainability of the project if the predicted market volume resulted to be an optimistic estimation and to have a lower charge on the users if the volume is higher.

## 8.7. Promotion

It has to be taking into account that the major barrier of the blockchain projects is the lack of active cryptocurrencies users. There are not reliable estimations of the total number of users, but ultimately the estimations are around 5 million of people. It has to be noted that platforms like this one don't target necessarily these users. But despite this still little user base of the cryptocurrencies the general awareness around them is pretty high, and more importantly it has increased greatly in the last year. To overcome this situation to kick-start this project the idea is to invite companies to try the platform and give them tokens to commission projects.

The main problem with this type of this token usage is that new tokens can devalue the token. In the case of this platform this will be mitigated by the fact that this wouldn't be a promotion where the tokens are simply being gifted. Instead, the final holders of this promotion tokens will be the users that decide to take on the projects commissioned by these companies. For them the token has the value of the work that they have performed, this avoids an instant sell of the tokens for quick process. Also, along this project it has been referred several times different systems that rely on the users holding tokens, as the arbitration fees, this gives another argument in favour of the gifted tokens ending in users' hands that hold them. Nevertheless, this can be further mitigated by designing the promotion as a loan to the company, and not as a gift. This will allow the companies interested in trying the platform to try the platform, and once their project is completed to pay back to the developers in fiat money, facilitating the adoption process. All the fiat money result of this loans will be reinverted in buying more promotion tokens, thus keeping a stable token value. But at the end, having a higher supply even if the price is kept stable has its cons, mainly that the potential increase of the token's value is decreased.

Other important factor to consider is the origin of this promotion tokens. Until now this origin has been mentioned simply to be tokens that were held by the developers. There are two different ways of creating these tokens. The most straight forward solution would be to create them at the same time that the rest of tokens, and hold them in reserve until the promotion of the platform starts. Another method is to mint them when the promotion of the platform starts. The first option has the advantage that the investors know how many tokens are being created before they invest their money, thus knowing the exact conditions that they are investing in. On the other hand, the second option has the advantage that at that point in time the exact need will be well known. Also, the price fluctuation of the token can mean that in the future, tokens can be valued at half or at ten times their ICO value.

A solution for this dilemma can be to implement a minting of tokens dependant on a community vote. Through smart contract is possible to implement voting rights to the token holders, giving them proportional voting rights respects to their holdings. This will allow to mint new tokens for promotion proposes in proposition of the development team. Then the community can decide if they accept such minting. This will have a balance because although the token holders don't want the maximum supply

to be raised they will seek the overall good for the project adoption, as this will be reflected in the token's value. Also, this solution can be combined with an initial token reserve for the developers, part as reward, as a help for determined tasks of the development that can be outsourced through the own platform and also as promotion.

# 9. Conclusion

In the following paragraphs a revision is going to be made out of the most important findings of this report. This project was conceived as a theoretical design of a decentralized platform, a research that would give as a result the needed material to create the whitepaper of the project. Creating a decentralize platform, given the complexity of the task, can be never fully outlined in its whitepaper. But this report shows an exhaustive research on how to implement the idea of a decentralized project exchange network and its possible features. Also, how these features meet the current technology and how this project meets the current state of the market.

From a platform design point of view, this report has outlined several features that enhance the platform from what can be expected from a standard platform. Decentralization provides a new paradigm in the security management of the user's data. This allows to minimize any security breaches and also provides an impartial demonstration of the intellectual property of a project, with a very low cost. Another advantage of using this decentralized approach is the low cost for the developers, that reduce their duties making possible to develop a platform with a more competitive charge to the users.

The technological challenges outlined in the objectives had been addressed deeply during the discussion of this report. For each challenge that was outlined, a detailed discussion of options and decisions were presented. The result of this discussion has been satisfactory. First the development of the platform was presented and the decision to develop an Ethereum ERC20 token was justified and detailed. The key features, of file sharing, file hash storage and arbitration had been designed and the technologies that power their functioning have been presented. Although some of these designs include solutions of compromise, such us implementing a complementary centralized file sharing application or controlling the referee reputation, the researched technologies allow to keep the project as decentralized as possible.

The economic analysis that has been carried out defines a strategy of funding and promotion, based in the blockchain technology. This strategy has been supported by real cases as well as a detailed explanation of the technologies and funding process to support it. As a result of the contrast between this strategy and other companies dedicated to online jobs define a promising future, both in the growth of online project exchange platforms and the popularity rise of the blockchain technology.

# 10. Future work

This project is heavily oriented upon the future work, as it a detailed design that helps the future development of the platform that I intend to carry out. In that sense the work has been satisfactory, as it has allowed me to have a deep understanding of the technologies and economics behind the project. This report, and the work that has been carried out, also serves a presentation card to the possible investors of the platform.

Another focus of the future work is to develop new ideas for the platform. Although during the report all of the ideas that I thought that would enhance the platform were presented and discussed, I think that the window for improvement can be still opened changing the focus of the platform. All of this report was directed to the Client/Creator exchange of a project, but given the possibilities of blockchain technologies the platform could be used with more objectives. Some ideas for the diversification of the platform into different proposes could be:

- Project Network could not only be used to develop project commissioned by individuals or companies. The platform could serve not only as a project management tool, but also as a crowdfunding resource. In which a community commissions a project and crowdfunds it.
- The platform could be used as a selling point for any online content, such as music, and the author could easily develop a smart contract to sell its production. The arbitration system could be used in this case to prevent any scams.

# 11. Potential clients covering letter

*This letter is based on the promotion method described in the section of the economic analysis. The objective of the letter is to offer a company the possibility to try the platform with tokens provided by the developers.*

Dear X,

I'm sure that as an executive of a big company, not only have you heard of Blockchain technology, but also you have wondered if there is any way to have an edge on your competitors by implementing it. This letter aims to present the platform Project Network, the opportunities that it presents and an offer to make a free trial of the service.

Project Network is a platform built on top of Ethereum. It brings together clients and creators interested in commissioning and completing all sorts of projects delivered in the form of a computer file. Powered by smart contracts the transactions of projects are handled in a secured way, immune to external intervention, where clients and creators can set objectives, deadlines and rewards. A double arbitration system is implemented to make the whole process more secure, and ensure that any malicious user is caught if they want to scam another user. All of the steps in the project development are handled by the platform, the commission of the project, the selection of the creator, the project file exchange and the payment. For more interesting data in our platform check out our white paper.

Our platform offers the possibility to your company to outsource all kind of tasks, with an emphasis in allowing a quick contact with creators and a fast resolution. Although this might be used to perform little tasks that overwhelm your workforce in certain moments it can be also used to outsource large projects that exceed the capacities of your company. You have a whole world of experts on the other side of the smart contract! And don´t look only to the project commissioning, your company may also find interesting opportunities to undertake with your experience and skills.

To try this new platform all you have to do is to contact back our development team. Thanks to the nature of our token, Projectcoin, new tokens are minted to give the opportunity to companies like yours to be the first to try our platform. If your project is completed satisfactory later you can pay back accordingly to your level of satisfaction.

Our team is waiting for hearing from you, to set your first project up and running.

Kind regards,
Project Network development team.

# References

1. *Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies.* **Tschorsch, F.** s.l. : IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC, 2016, Vols. 18 Issue: 3 Pages: 2084-2123.

2. *Blockchain beyond bitcoin.* **Underwood, Sarah.** 11, 2016, Vol. 59 .

3. **Juniper Research: Blockchain Enterprise Survey August 2017. [Online] 31 July 2017. https://www.juniperresearch.com/resources/infographics/blockchain-enterprise-survey-august-2017.**

**4. *Overview of business innovations and research opportunities in blockchain and introduction to the special issue.* Zhao, J. Leon. 2016, Vols. 2, pg 28.**

**5. *Bitcoin: A Peer-to-Peer Electronic Cash System.* Nakamoto, Satoshi. 2008.**

**6. Landman, Nathan. Secure Hashing Algorithms. [Online] 15 Oct 20017. https://brilliant.org/wiki/secure-hashing-algorithms/.**

**7. Descriptions of SHA-256, SHA-384, and SHA-512. [Online] [Cited: 30 Oct 2017.] http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf.**

**8. Cryptocompare. [Online] 28 Sep 2017 . https://www.cryptocompare.com/wallets/guides/how-do-digital-signatures-in-bitcoin-work/.**

**9. CuriousInventor. How Bitcoin Works in 5 Minutes (Technical). [Online] 16 Oct 2017. https://www.youtube.com/watch?v=l9jOJk30eQs.**

**10. ¿Qué es la Cadena de Bloques (Blockchain)? [Online] 18 Oct 2017. http://blog.bit2me.com/es/que-es-cadena-de-bloques-blockchain/.**

**11. Swan, Melanie. *Blockchain: Blueprint for a New Economy.* 20015.**

**12. *A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM.* Buterin, Vitalik. 2013.**

**13. *An Introduction to Ethereum and Smart Contracts: a Programmable Blockchain.* Peyrott, Sebastián. 2017.**

**14. *Factom whitepaper.* Snow, Paul. 2014.**

**15. *Blockchain Technology as an Institution of Property.* Ishmaev, G. 5, 2017, Vol. 48.**

**16. *FistBlood: A Decentralized eSports Platform Based on Smart Contracts.* Cuesta, Marco. 2016.**

**17.** ***Lisk whitepaper.*** **Kordek, Max. 2016.**

**18.** ***Neo: A distributed network for the Smart Economy.*** **Hongfei, Da. 2016.**

**19.** ***A Measurement Study of Peer-to-Peer File Sharing.*** **Saroiu, Stefan. 2002, Proceedings Volume 4673, Multimedia Computing and Networking 2002.**

**20. takeafile. [Online] 25 Oct 2017. https://takeafile.com/es/.**

**21. reep.io. [Online] 25 Oct 2017. https://reep.io/.**

**22. Send Anywhere. [Online] 2017 Oct 2017. https://send-anywhere.com/.**

**23. Lee, Wallis. Bitcoin price since inception (logarithmic scale). [Online] 27 Oct 2017. https://www.reddit.com/r/Bitcoin/comments/6tlkie/bitcoin_price_since_inception_logarithmic_scale/?st=j9ekvnj2&sh=cf2f52c8.**

**24. Cryptocurrency Market Capitalizations. [Online] 25 Oct 2017. https://coinmarketcap.com/.**

**25. Kharpal, Arjun. CNBC. [Online] 9 Aug 2017. https://www.cnbc.com/2017/08/09/initial-coin-offerings-surpass-early-stage-venture-capital-funding.html.**

**26.** ***Cryptocurrency: A New Investment Opportunity?*** **Chuen, David Lee Kuo. 2017.**

**27.** ***Exploring signals for investing in an Initial Coin Offering (ICO).*** **Yadav, Mohit. 2017.**

**28.** ***Initial Coin Offering (ICO) Risk, Value and Cost in Blockchain Trustless Crypto Markets.*** **Venegas, Percy. 2017.**

**29.** ***Cointelegraph.*** **[Online] https://cointelegraph.com/news/coinbase-paves-way-for-new-altcoin-support-with-erc20-upgrade.**

**30. Ethereum. [Online] https://www.ethereum.org/crowdsale.**

**31.** ***Ether reseach.*** **[Online] https://ethresear.ch/t/explanation-of-daicos/465.**

**32.** ***Coinschedule.*** **[Online] https://www.coinschedule.com/.**

**33.** ***Behind the enemy lines.*** **[Online] http://www.behind-the-enemy-lines.com/2012/11/is-mechanical-turk-10-billion-dollar.html.**

**34.** ***Techcrunch.*** **[Online] https://techcrunch.com/2015/11/12/fiverr-ceo-on-raising-60-million-in-fresh-funding-its-a-land-grab-right-now/.**

**35.** ***Blockchain: understanding the potential.*** **Taylor, Simon. 2015.**

**36. *mbopartners.* [Online] https://www.mbopartners.com/uploads/files/state-of-independence-reports/StateofIndependence-2017-Final.pdf.**

# Appendix: Project exchange smart contract

```
pragma solidity ^0.4.18;
contract exchange{
    address public client;
    address public creator;
    bool public cancelcontract=false;
    bool public callarbitration=false;
    uint public payment;
    uint public fee;
    uint public b=this.balance;

    function exchange(uint newpayment, uint newfee) public{
        client= msg.sender;
        payment=newpayment;
        fee=newfee;
    }
    modifier ifclient(){
        if(client!=msg.sender){
            revert();
        }
        _;
    }
    modifier iffunds(){
        if(this.balance==0){
            revert();
        }
        _;
    }
    modifier ifcreator(){
        if(creator!=msg.sender){
            revert();
        }
        _;
```

```
    }
    modifier ifcancelcontract(){
        if(!cancelcontract){
            revert();
        }
        _;
    }
    modifier ifarbitration(){
        if(!callarbitration){
            revert();
        }
        _;
    }
    modifier ifpayment(){
        if(msg.value != payment+fee){
            revert();
        }
        _;
    }
    modifier iffee(){
        if(msg.value !=fee){
            revert();
        }
        _;
    }
    function client_funding() payable ifclient ifpayment public {}
    function creator_funding() payable ifcreator iffee public {}
    function getfunds() constant public returns(uint) {
        return this.balance;
    }
    function setcreator(address newcreator) public ifclient iffunds {
        creator= newcreator;
    }
    function clientveredict(bool clientjudgment) public ifclient{
```

```
        if(clientjudgment){
            creator.transfer(this.balance);
        }
        else{
            cancelcontract=true;
        }
    }
    function creatorveredict(bool creatorjudgment) public ifcreator ifcancelcontract{
        if(creatorjudgment){
            client.transfer(this.balance);
        }
        else{
            callarbitration=true;
        }
    }
    function arbitration(bool arbitrationjudgment) public ifarbitration {
        if(arbitrationjudgment){
            client.transfer(this.balance);
        }
        else{
            creator.transfer(this.balance);
        }
    }
}
```

# Appendix: Token creation smart contract

The following smart contract is the smart contract responsible of the creation of the FirstBlood token. It follows the ERC20 protocol and also is responsible for the ICO of the token. The smart contract can be found in the Ethereum blockchain, and its address is:

**0xAf30D2a7E90d7DC361c8C4585e9BB7D2F6f15bc7**

```
/**
 * Overflow aware uint math functions.
 *
 * Inspired by https://github.com/MakerDAO/maker-otc/blob/master/contracts/simple_market.sol
 */
contract SafeMath {
  //internals
  function safeMul(uint a, uint b) internal returns (uint) {
    uint c = a * b;
    assert(a == 0 || c / a == b);
    return c;
  }
  function safeSub(uint a, uint b) internal returns (uint) {
    assert(b <= a);
    return a - b;
  }
  function safeAdd(uint a, uint b) internal returns (uint) {
    uint c = a + b;
    assert(c>=a && c>=b);
    return c;
  }
  function assert(bool assertion) internal {
    if (!assertion) throw;
  }
}
/**
 * ERC 20 token
 *
```

```
 * https://github.com/ethereum/EIPs/issues/20
 */
contract Token {
    /// @return total amount of tokens
    function totalSupply() constant returns (uint256 supply) {}
    /// @param _owner The address from which the balance will be retrieved
    /// @return The balance
    function balanceOf(address _owner) constant returns (uint256 balance) {}
    /// @notice send `_value` token to `_to` from `msg.sender`
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transfer(address _to, uint256 _value) returns (bool success) {}
    /// @notice send `_value` token to `_to` from `_from` on the condition it is approved by `_from`
    /// @param _from The address of the sender
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {}

    /// @notice `msg.sender` approves `_addr` to spend `_value` tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @param _value The amount of wei to be approved for transfer
    /// @return Whether the approval was successful or not
    function approve(address _spender, uint256 _value) returns (bool success) {}
    /// @param _owner The address of the account owning tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @return Amount of remaining tokens allowed to spent
    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {}
    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256 _value);
}
/**
 * ERC 20 token
```

```
 *
 * https://github.com/ethereum/EIPs/issues/20
 */
contract StandardToken is Token {
    /**
     * Reviewed:
     * - Interger overflow = OK, checked
     */
    function transfer(address _to, uint256 _value) returns (bool success) {
        //Default assumes totalSupply can't be over max (2^256 - 1).
        //If your token leaves out totalSupply and can issue more tokens as time goes on, you need to check if it doesn't wrap.
        //Replace the if with this one instead.
        if (balances[msg.sender] >= _value && balances[_to] + _value > balances[_to]) {
        //if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }
    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
        //same as above. Replace this line with the following if you want to protect against wrapping uints.
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && balances[_to] + _value > balances[_to]) {
        //if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }
    function balanceOf(address _owner) constant returns (uint256 balance) {
```

```
        return balances[_owner];
    }
    function approve(address _spender, uint256 _value) returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
        return true;
    }
    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {
      return allowed[_owner][_spender];
    }
    mapping(address => uint256) balances;
    mapping (address => mapping (address => uint256)) allowed;
    uint256 public totalSupply;
}
/**
 * First blood crowdsale crowdsale contract.
 *
 * Security criteria evaluated against http://ethereum.stackexchange.com/questions/8551/methodological-security-review-of-a-smart-contract
 *
 *
 */
contract FirstBloodToken is StandardToken, SafeMath {
    string public name = "FirstBlood Token";
    string public symbol = "1ST";
    uint public decimals = 18;
    uint public startBlock; //crowdsale start block (set in constructor)
    uint public endBlock; //crowdsale end block (set in constructor)

    // Initial founder address (set in constructor)
    // All deposited ETH will be instantly forwarded to this address.
    // Address is a multisig wallet.
    address public founder = 0x0;
```

```
    // signer address (for clickwrap agreement)
    // see function() {} for comments
    address public signer = 0x0;
    uint public etherCap = 465313 * 10**18; //max amount raised during crowdsale (5.5M USD worth of ether will be measured with a moving average market price at beginning of the crowdsale)
    uint public transferLockup = 370285; //transfers are locked for this many blocks after endBlock (assuming 14 second blocks, this is 2 months)
    uint public founderLockup = 2252571; //founder allocation cannot be created until this many blocks after endBlock (assuming 14 second blocks, this is 1 year)
    uint public bountyAllocation = 2500000 * 10**18; //2.5M tokens allocated post-crowdsale for the bounty fund
    uint public ecosystemAllocation = 5 * 10**16; //5% of token supply allocated post-crowdsale for the ecosystem fund
    uint public founderAllocation = 10 * 10**16; //10% of token supply allocated post-crowdsale for the founder allocation
    bool public bountyAllocated = false; //this will change to true when the bounty fund is allocated
    bool public ecosystemAllocated = false; //this will change to true when the ecosystem fund is allocated
    bool public founderAllocated = false; //this will change to true when the founder fund is allocated
    uint public presaleTokenSupply = 0; //this will keep track of the token supply created during the crowdsale
    uint public presaleEtherRaised = 0; //this will keep track of the Ether raised during the crowdsale
    bool public halted = false; //the founder address can set this to true to halt the crowdsale due to emergency
    event Buy(address indexed sender, uint eth, uint fbt);
    event Withdraw(address indexed sender, address to, uint eth);
    event AllocateFounderTokens(address indexed sender);
    event AllocateBountyAndEcosystemTokens(address indexed sender);
    function FirstBloodToken(address founderInput, address signerInput, uint startBlockInput, uint endBlockInput) {
        founder = founderInput;
        signer = signerInput;
        startBlock = startBlockInput;
        endBlock = endBlockInput;
    }
    /**
     * Security review
```

```
    *
    * - Integer overflow: does not apply, blocknumber can't grow that high
    * - Division is the last operation and constant, should not cause issues
    * - Price function plotted https://github.com/Firstbloodio/token/issues/2
    */
   function price() constant returns(uint) {
      if (block.number>=startBlock && block.number<startBlock+250) return 170; //power hour
      if (block.number<startBlock || block.number>endBlock) return 100; //default price
      return 100 + 4*(endBlock - block.number)/(endBlock - startBlock + 1)*67/4; //crowdsale price
   }
   // price() exposed for unit tests
   function testPrice(uint blockNumber) constant returns(uint) {
      if (blockNumber>=startBlock && blockNumber<startBlock+250) return 170; //power hour
      if (blockNumber<startBlock || blockNumber>endBlock) return 100; //default price
      return 100 + 4*(endBlock - blockNumber)/(endBlock - startBlock + 1)*67/4; //crowdsale price
   }
   // Buy entry point
   function buy(uint8 v, bytes32 r, bytes32 s) {
      buyRecipient(msg.sender, v, r, s);
   }
   /**
    * Main token buy function.
    *
    * Security review
    *
    * - Integer math: ok - using SafeMath
    *
    * - halt flag added - ok
    *
    * Applicable tests:
    *
    * - Test halting, buying, and failing
    * - Test buying on behalf of a recipient
    * - Test buy
```

```
    * - Test unhalting, buying, and succeeding
    * - Test buying after the sale ends
    *
    */
   function buyRecipient(address recipient, uint8 v, bytes32 r, bytes32 s) {
      bytes32 hash = sha256(msg.sender);
      if (ecrecover(hash,v,r,s) != signer) throw;
      if (block.number<startBlock || block.number>endBlock || safeAdd(presaleEtherRaised,msg.value)>etherCap || halted) throw;
      uint tokens = safeMul(msg.value, price());
      balances[recipient] = safeAdd(balances[recipient], tokens);
      totalSupply = safeAdd(totalSupply, tokens);
      presaleEtherRaised = safeAdd(presaleEtherRaised, msg.value);

      if (!founder.call.value(msg.value)()) throw; //immediately send Ether to founder address

      Buy(recipient, msg.value, tokens);
   }

   /**
    * Set up founder address token balance.
    *
    * allocateBountyAndEcosystemTokens() must be calld first.
    *
    * Security review
    *
    * - Integer math: ok - only called once with fixed parameters
    *
    * Applicable tests:
    *
    * - Test bounty and ecosystem allocation
    * - Test bounty and ecosystem allocation twice
    *
    */
```

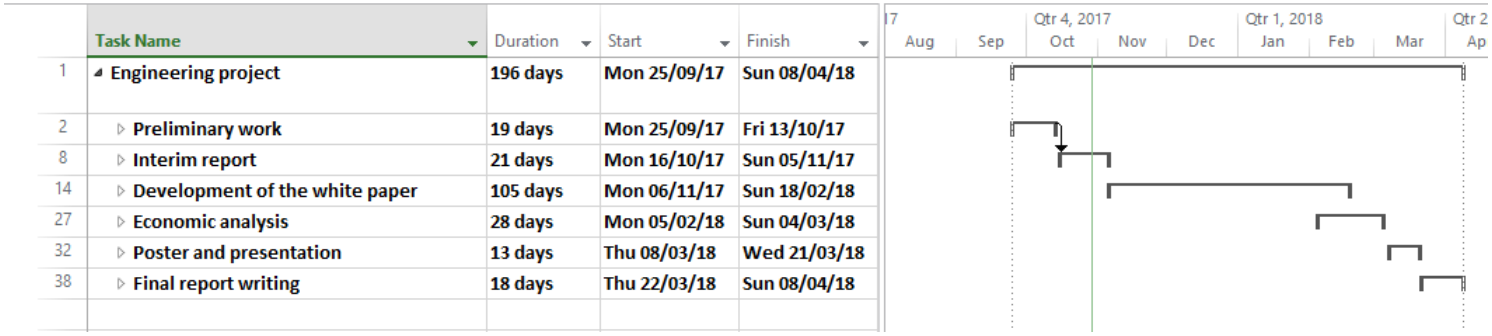
```
    function allocateFounderTokens() {
        if (msg.sender!=founder) throw;
        if (block.number <= endBlock + founderLockup) throw;
        if (founderAllocated) throw;
        if (!bountyAllocated || !ecosystemAllocated) throw;
        balances[founder] = safeAdd(balances[founder], presaleTokenSupply * founderAllocation / (1 ether));
        totalSupply = safeAdd(totalSupply, presaleTokenSupply * founderAllocation / (1 ether));
        founderAllocated = true;
        AllocateFounderTokens(msg.sender);
    }
    /**
     * Set up founder address token balance.
     *
     * Set up bounty pool.
     *
     * Security review
     *
     * - Integer math: ok - only called once with fixed parameters
     *
     * Applicable tests:
     *
     * - Test founder token allocation too early
     * - Test founder token allocation on time
     * - Test founder token allocation twice
     *
     */
    function allocateBountyAndEcosystemTokens() {
        if (msg.sender!=founder) throw;
        if (block.number <= endBlock) throw;
        if (bountyAllocated || ecosystemAllocated) throw;
        presaleTokenSupply = totalSupply;
        balances[founder] = safeAdd(balances[founder], presaleTokenSupply * ecosystemAllocation / (1 ether));
```

```
        totalSupply = safeAdd(totalSupply, presaleTokenSupply * ecosystemAllocation / (1 ether));
        balances[founder] = safeAdd(balances[founder], bountyAllocation);
        totalSupply = safeAdd(totalSupply, bountyAllocation);
        bountyAllocated = true;
        ecosystemAllocated = true;
        AllocateBountyAndEcosystemTokens(msg.sender);
    }
    /**
     * Emergency Stop crowdsale.
     *
     *  Applicable tests:
     *
     * - Test unhalting, buying, and succeeding
     */
    function halt() {
        if (msg.sender!=founder) throw;
        halted = true;
    }
    function unhalt() {
        if (msg.sender!=founder) throw;
        halted = false;
    }
    /**
     * Change founder address (where crowdsale ETH is being forwarded).
     *
     * Applicable tests:
     *
     * - Test founder change by hacker
     * - Test founder change
     * - Test founder token allocation twice
     *
     */
    function changeFounder(address newFounder) {
        if (msg.sender!=founder) throw;
```

```
    founder = newFounder;
  }
  /**
   * ERC 20 Standard Token interface transfer function
   *
   * Prevent transfers until freeze period is over.
   *
   * Applicable tests:
   *
   * - Test restricted early transfer
   * - Test transfer after restricted period
   */
  function transfer(address _to, uint256 _value) returns (bool success) {
    if (block.number <= endBlock + transferLockup && msg.sender!=founder) throw;
    return super.transfer(_to, _value);
  }
  /**
   * ERC 20 Standard Token interface transfer function
   *
   * Prevent transfers until freeze period is over.
   */
  function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
    if (block.number <= endBlock + transferLockup && msg.sender!=founder) throw;
    return super.transferFrom(_from, _to, _value);
  }
  /**
   * Do not allow direct deposits.
   *
   * All crowdsale depositors must have read the legal agreement.
   * This is confirmed by having them signing the terms of service on the website.
   * They give their crowdsale Ethereum source address on the website.
   * Website signs this address using crowdsale private key (different from founders key).
   * buy() takes this signature as input and rejects all deposits that do not have
   * signature you receive after reading terms of service.
```

```
    *
    */
   function() {
      throw;
   }
}
```

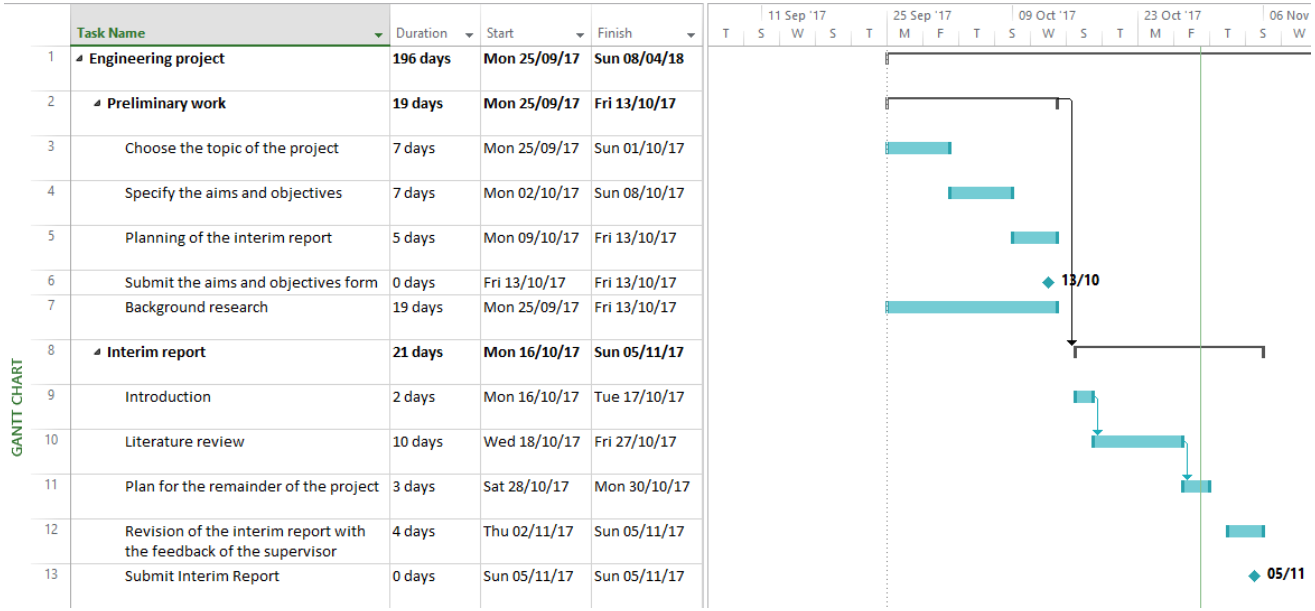
# Appendix: Gantt diagram



| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Engineering project** | **196 days** | **Mon 25/09/17** | **Sun 08/04/18** |
| 2 | **Preliminary work** | **19 days** | **Mon 25/09/17** | **Fri 13/10/17** |
| 8 | **Interim report** | **21 days** | **Mon 16/10/17** | **Sun 05/11/17** |
| 14 | **Development of the white paper** | **105 days** | **Mon 06/11/17** | **Sun 18/02/18** |
| 27 | **Economic analysis** | **28 days** | **Mon 05/02/18** | **Sun 04/03/18** |
| 32 | **Poster and presentation** | **13 days** | **Thu 08/03/18** | **Wed 21/03/18** |
| 38 | **Final report writing** | **18 days** | **Thu 22/03/18** | **Sun 08/04/18** |



*Gannt diagram 1*



GANTT CHART

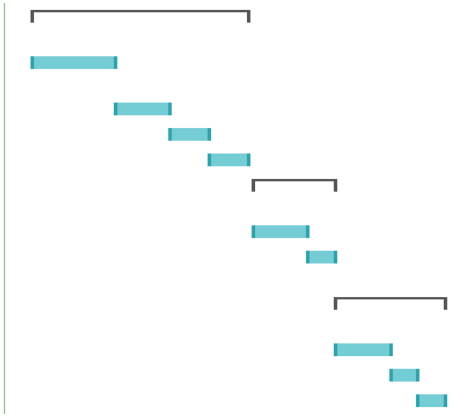
| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Engineering project** | **196 days** | **Mon 25/09/17** | **Sun 08/04/18** |
| 2 | **Preliminary work** | **19 days** | **Mon 25/09/17** | **Fri 13/10/17** |
| 3 | Choose the topic of the project | 7 days | Mon 25/09/17 | Sun 01/10/17 |
| 4 | Specify the aims and objectives | 7 days | Mon 02/10/17 | Sun 08/10/17 |
| 5 | Planning of the interim report | 5 days | Mon 09/10/17 | Fri 13/10/17 |
| 6 | Submit the aims and objectives form | 0 days | Fri 13/10/17 | Fri 13/10/17 |
| 7 | Background research | 19 days | Mon 25/09/17 | Fri 13/10/17 |
| 8 | **Interim report** | **21 days** | **Mon 16/10/17** | **Sun 05/11/17** |
| 9 | Introduction | 2 days | Mon 16/10/17 | Tue 17/10/17 |
| 10 | Literature review | 10 days | Wed 18/10/17 | Fri 27/10/17 |
| 11 | Plan for the remainder of the project | 3 days | Sat 28/10/17 | Mon 30/10/17 |
| 12 | Revision of the interim report with the feedback of the supervisor | 4 days | Thu 02/11/17 | Sun 05/11/17 |
| 13 | Submit Interim Report | 0 days | Sun 05/11/17 | Sun 05/11/17 |



*Gannt diagram 2*

GANTT CHART

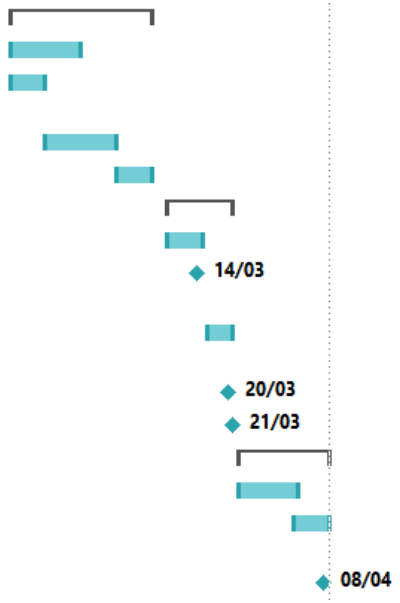
| | | | | |
|---|---|---|---|---|
| 15 | **Blockchain development** | **55 days** | **Mon 06/11/17** | **Sat 30/12/17** |
| 16 | Bases of the development of the decentralized app | 21 days | Mon 06/11/17 | Sun 26/11/17 |
| 17 | Smart contract functioning | 14 days | Mon 27/11/17 | Sun 10/12/17 |
| 18 | Record file transactions | 10 days | Mon 11/12/17 | Wed 20/12/17 |
| 19 | Arbitration system | 10 days | Thu 21/12/17 | Sat 30/12/17 |
| 20 | **File exchange system** | **21 days** | **Mon 01/01/18** | **Sun 21/01/18** |
| 21 | File exchange principles | 14 days | Mon 01/01/18 | Sun 14/01/18 |
| 22 | Integrating file exchanges with the rest of the platform | 7 days | Mon 15/01/18 | Sun 21/01/18 |
| 23 | **Usage conditions of the platform** | **28 days** | **Mon 22/01/18** | **Sun 18/02/18** |
| 24 | Specify the arbitration system | 14 days | Mon 22/01/18 | Sun 04/02/18 |
| 25 | Fees in the platform | 7 days | Mon 05/02/18 | Sun 11/02/18 |
| 26 | Term conditions | 7 days | Mon 12/02/18 | Sun 18/02/18 |



*Gannt diagram 3*

GANTT CHART

| | | | | |
|---|---|---|---|---|
| 27 | **Economic analysis** | **28 days** | **Mon 05/02/18** | **Sun 04/03/18** |
| 28 | Background research | 14 days | Mon 05/02/18 | Sun 18/02/18 |
| 29 | Choose the mathematical models that will be used | 7 days | Mon 05/02/18 | Sun 11/02/18 |
| 30 | Apply the mathematical models | 14 days | Mon 12/02/18 | Sun 25/02/18 |
| 31 | Summarize the conclusions | 7 days | Mon 26/02/18 | Sun 04/03/18 |
| 32 | **Poster and presentation** | **13 days** | **Thu 08/03/18** | **Wed 21/03/18** |
| 33 | Poster preparation | 7 days | Thu 08/03/18 | Wed 14/03/18 |
| 34 | Submit the draft for the supervisor | 0 days | Wed 14/03/18 | Wed 14/03/18 |
| 35 | Presentation preparation | 5 days | Fri 16/03/18 | Tue 20/03/18 |
| 36 | Final poster submission | 0 days | Tue 20/03/18 | Tue 20/03/18 |
| 37 | Poster presentation | 0 days | Wed 21/03/18 | Wed 21/03/18 |
| 38 | **Final report writing** | **18 days** | **Thu 22/03/18** | **Sun 08/04/18** |
| 39 | Integrate the previous work | 12 days | Thu 22/03/18 | Mon 02/04/18 |
| 40 | Revision of the interim report with the feedback of the supervisor | 7 days | Mon 02/04/18 | Sun 08/04/18 |
| 41 | Submit the final report | 0 days | Sun 08/04/18 | Sun 08/04/18 |




*Gannt diagram 4*