INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Ethereum Smart Contracts for Educational Certificates

EMANUEL FILIPE GOMES DIAS
outubro de 2018

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Ethereum Smart Contracts for Educational Certificates

Emanuel Filipe Gomes Dias

Dissertation to obtain the Master's Degree in Informatics, Area of Expertise in Knowledge and Information Systems

Advisor: Isabel de Fátima Silva Azevedo

Porto, October 2018

Acknowledgments

I'd like to express all my gratitude to my advisor, Professor Isabel Azevedo, for the guidance and motivation that made this thesis project possible in being accomplished.

To my parents for giving me the indispensable education, support and determination to surpass every obstacle that my life has to offer.

And lastly to my family and friends for all the support provided throughout of these years.

Abstract

Since blockchains started making its steps for recognition to the world, it began achieving new forms of entries in the daily life, a simple example is the way society trade with virtual coins - cryptocurrency.

By this definition, the education fields can take advantage of this flexible system to ensure the recognition's work of the scholar. With the ability of students getting credit for the knowledge that happens anywhere, not just in schools or formal classes, to be certificated in the blockchain so it answers all sorts of manners of availability and validation.

This work is prompted to demonstrate how the smart contracts transactions can be used in learning areas, to historically maintain educational certified documents on the blockchain.

In such way, the investigation of the Ethereum's blockchain is taken into consideration, to obtain an essential overview of the functionalities that allow to create a prototype, for the certificate management between entities.

Keywords: Ethereum, Blockchain, Smart Contracts, Education, Certificate

Resumo

Desde que os blockchains começaram a tomar os seus passos no reconhecimento mundial, foi possível contrastar novas mudanças na vida diária, um exemplo simples é a forma como a sociedade troca valores com moedas virtuais - cryptocurrency.

Por essa afirmação, áreas da educação podem aproveitar esse sistema flexível para garantir o reconhecimento do trabalho acadêmico do aluno. Com a capacidade de os alunos obterem reconhecimento pela aprendizagem que se sucede em qualquer lugar, não apenas nas escolas ou nas aulas, de modo a que seja assegurado no blockchain, comprovando vários tipos de disponibilidade e validação.

Este trabalho demonstra como as transações de smart contracts podem ser usadas nas áreas da educação, mantendo historicamente os documentos certificados no blockchain.

Desta forma, a investigação do blockchain do Ethereum tida em consideração, para obter uma visão essencial das funcionalidades que permitem desenvolver um protótipo, para a gestão dos certificados entre as entidades.

Palavras-chave: Ethereum, Blockchain, Smart Contracts, Educação, Certificados

Table of Contents

List of Images

List of Tables

Acronyms

AHP	Analytic Hierarchy Process
API	Application Programming Interface
BPMN	Business Process Model and Notation
DApp	Decentralized Application
EIP	Ethereum Improvement Proposal
EVM	Ethereum Virtual Machine
ICO	Initial Coin Offering
IDE	Integrated Development Environment
IPFS	InterPlanetary File System
NCD	New Concept Development
P2P	Peer-to-Peer
PoS	Proof-of-Stake
PoW	Proof-of-Work
QFD	Quality Function Deployment
ROI	Return on Investment
UML	Unified Modeling Language
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
SHA	Secure Hash Algorithm

1 Introduction

An introduction to this work is ensured in this chapter, to completely understand the context, problem, objectives, approach, experiments, and evaluation. All of this is engaged to give the reader an summary of the project's intentions.

1.1 Context

In recent years, the society has proven to receive a special attention for the demand of greater transparency and reliability in the public administration, and the expansion models for a technology capable to provide a powerful sense of revolution. Blockchains - in which can provide both needs (D. Tapscott & A. Tapscott, 2017 [a]).

An application that involves around the blockchain network, in the learning services, can consist to serve the main purpose of accessibility and verification on the obtained knowledge, answering questions on how to trust the information students provide to assessment organizations, educational institutions, and employers.

Since information is difficult to verify, it is likely to be manipulated to benefit different interests at various times. Even so, the information is deemed to be reliable where it is centralized and stored by established authorities, who can at any time violate and manipulate the data.

These are the main reasons that the community demands a greater transparency and reliability in the information. By that statement, the global interest in blockchain technology has been rising, creating a scenario with great potential for the development of new applications.

Considering smart contracts (Linklaters, 2017), cryptocurrencies, distributed consensus and all the technology that is available, these methods alone are surely defining and revolutionizing the way of authenticating the activities and transactions, in the stored information. All this information that lies within an alternative form of data management, the Blockchain, shapes a

decentralized trust relationship, whereas there isn't any trust to anyone for the verification of the transactions. Thus, this trust comes by default in the way that the network was designed.

Companies recognize the value of transparency in knowledge - this is becoming progressively clear that is a fundamental shift in the way of analyzing the intangible assets of data, in the knowledge-based sectors, exclusively in the human resources activity (S. Taylor, 2015).

1.2 Problem

For a long time, issued educational paper documents has been the support of knowledge: from preschool certificates to high school, and higher educational diplomas (R. Dore, 1997), and, also, by the represented university's diplomas or certificates. The amount of knowledge acquired by individuals is measured on a piece of paper, named as diploma/certificate.

The quality inferred by the reputation and prestige of the issued certificate's form, has several faults. Often the unavailability of the issuing real certificates and impartial forms cannot verify that someone has the qualification, proven by the certificate, that truly attained that level of knowledge (or primary points of that knowledge). For instance, for a group of people who graduated in the same year and all have the same qualification, the usage of evaluating tests might be the only way to know who has learned the reliable and necessary knowledge for the companies recognize their capabilities (E. Pollard, et al., 2015). That can prove to be a long and heavy task, to the simple end of understanding if the candidate has those qualities shown in the document.

Fake academic certificates and other credentials are a huge problem (J. Southurst, 2016), as they are issued in a traditional way, simply making copies and prints without any control that guarantees the integrity and originality of the document.

Meanwhile, there are digital counterfeit certifications being made (G. Gollin, 2008) and progressively is easier to forge than paper certificates, contracting companies have trust issues relying on the digital diplomas shown in the interview, even from reliable digital sources that are proven to be unknown to the company. This might force the candidates to resort to paper certificates to ensure the conviction of those certificates.

Most problems in showing certification documents involve the authentication and accessibility of such educational certifications. It is comprehensive to take careful execution when tackling these adversities, since the documents are utmost important to express all qualifications the person has obtained, demonstrating all the acquired qualifications to various contracting organizations.

1.3 Objectives

The aim of this dissertation is to study and explore the blockchain Ethereum and comprehend its usefulness for implementing a prototype solution for the storage and verification of educational certificates.

An understanding of the blockchain potential undergoes a phase where all the required information for the project's context is collected. Allows getting the knowledge required in the context standpoint, for example, significant blockchain information, the Ethereum platform, and the educational certificates. After this stage is documented, the development of the prototype shall be created based on the obtained information.

This thesis prototype must allow the Learners to view a list of the assigned certificates, acquired from Educational Institutions. The intention of the certificates' list, associated to a Learner account, is to show them to the Contracting Organizations, with the simplicity and veracity attained from the application and the blockchain technology.

On the other hand, Administrators on the application has the responsibility in the maintenance of the prototype reliability. With the permission to insert and modify records on the prototype database, to update with the authorized Educational Institutions that can issue Certificates inside the prototype application.

Similar fully developed projects are the Blockcerts (MIT Institution, 2016) and TrueRec (TrueRec, 2017), where the certificates/diplomas are stored in a blockchain, to preserve the earned achievements in the person's account. Both applications have the same purpose: issue, manage and display the learning achievements, in which is useful to understand these projects to analyze the best approach of the requirements and development methods.

The prototype development also the same purpose as those developed projects, but with additional intention to analyze the best possible application of educational certificates in an Ethereum blockchain network: if in a closed private network or in a main public network.

1.4 Approach and Development Process

Design Science Research (DSR) methodology was used with the purpose of obtaining a general solution concept (A. Henver, et al., 2014). This methodology is composed of seven guidelines[1], and it is better concise by three main guidelines, according to A. Henver[2].

The three main guidelines are the definition and explanation of the project's Artifact, Research Contribution, and Evidences. Each guideline is better detailed in some chapters, as follows:

- **Artifact -** representation of the process.

  The related chapter 1 Introduction gives an overview to the intended Artifact and chapter 2 State of Art, clarifies important general aspects of the Artifact.

- **Research Contribution -** contribution of knowledge in the information system field.

  Related to the chapter 3 Value Analysis, that adjusts the market perspective to the project's idea, and chapters 6 Tests and Experimentations and 7 Conclusion delivers the most relevant tests in the prototype, that results in knowledge contribution for the information system.

- **Evidences -** problem solving and innovation of the project.

  Chapters that are associated with this main guideline are chapters 4 Requirements Analysis, 5 Implementation and 6 Tests and Experimentations. Entirely describes the implemented prototype as a solution to the specified problem, with the interpretation of the results in the tests and experimentations.

The development process is designed by the creation of a private network using Ethereum blockchain, allowing to perform the necessary steps to construct a simple and controlled prototype in an empty blockchain network. This also enables the usage of the Ethereum without any cost in transactions exchange, during testing.

Since this technology is based on the blockchain, to maintain the validation of the network, exploration must be engaged by miners[3]. For this matter, the creation of entities as miners are expected to be added to the network, since without any the transactions cannot be verified and added to the blockchain.

An application to truly make this process effortless and perceptible is kept in consideration during the implementation so that the application user can issue, edit and view all the information required in their account.

To fully accomplish all the requirements in the developed project, tests must be conducted so it can be identified the end results, to consider the prototype's performance in real environment.

[1] DSR's seven guidelines are represented in the Annex A - Design Science Research Guidelines.
[2] An interview to A. Hevner discussing DSR methodology (A. Hevner, 2015).
[3] Entities that represent processes to validate the transactions and blocks, in order to insert them into the blockchain network and receive the fee cost rewards.

This project principal consideration is to send certificate's information, as a smart contract, to the blockchain network, in which a hypothesis must be around in such consideration. The general hypothesis are the fee costs in the issue of such contracts and the time occurred between the moment that the educational institution issues the contract, as a transaction, and the moment it is validated to the blockchain.

The costs are presented in the main public blockchain network[4] and can prove an evaluation method to determine the costs that institutions must maintain, to the contracts being continually issued to learners. The validation time is essential to understand how much time it takes for such contract be present in the learner's account.

These considerations can only be accomplished in a testing blockchain network (since the real blockchain network is expensive). This contributes to the consequence that no "real" cost values are evaluated; however, these testing values ensure a possible speculation of the interaction within the blockchain to the validation of the inserted contracts. In later chapters, these experimentations shall be explained in more detail, with the usage of the Ropsten[5] blockchain testing network.

1.5 Document Structure

The document is structured in the following chapters:

- **Introduction:** makes a brief introduction to the project
- **State of Art:** explains in detail the expressions used in this document
- **Value Analysis:** uncovers the value overview
- **Requirements Analysis:** explanation of the prototype approach
- **Implementation:** detail of procedures taken in the development
- **Tests and Experimentations:** exposes and examines prototype's smart contract results
- **Conclusion:** encloses the document with some thoughts regarding the theme
- **Annexes:** composes of sections that contain all the used annexes in the dissertation

[4] The main/real blockchain network is a public blockchain network in which real money circulation is held and transfer to other accounts.
[5] Ropsten is a public blockchain network in which users can deploy transactions without any real cost associated, for testing purposes.

2 State of Art

This chapter focus on the essential knowledge regarding this document theme. A chapter that is useful to understand a few important details before advancing to the succeeding chapters, describing the developed solution.

2.1 Certifications and Diplomas

Learning and teaching processes, used in certified documents, are held as a system of historical and analytic use, to establish a concrete appreciation of the student's learning (S. Shin, 2012).

This conception acknowledges the school subjects that are responsible to capture the information made in the documents for historical purposes, allowing one to understand them. This aims to understand the approaches, the reasons and the methods engaged in the education course, in summary, to acknowledge the potential learning assumed by the person.

The contents of teaching, organized in the different areas of knowledge and subjects, are documented in educational certifications or diplomas, and in the elaborated integrated curriculum (possibly composes of various certifications). In this way, it will be seen that general and professional knowledge are only distinguished in the historical record of such documents.

Receiving a certificate composes a great personal achievement for many, regardless of the course, workload, and method of study. The received document enables to show all the effort, dedication learned and mastered in a subject, either in paper or digitally.

Paper certification is tangible and responsible for involving the printed information to the person' signature. As for in digital documents is composed only by the representation of data, as well there is no simple way to relate the document to the signature, which requires a computer to view and confirm (L. M. Singer & P. A. Alexander, 2017). Among the mandatory

fields of the digital certificate are the identification and signature of the issuing entity, which allows verification of the authenticity and integrity of the certificate.

The similarity of the digital signature and the handwritten signature is restricted to the principle of assigning authorship to a document. In the manuscript, the signatures follow a pattern, possessing personal characteristics of each individual (D. P. Franco, et al., 2013). The veracity of the handwritten signature is made by a visual comparison to a true signature, such as that of the official identity document.

Despite the differences, the digital signature technique is an effective way to ensure the authorship of electronic documents. The legal validity of electronic documents and the use of digital certificates always assign authenticity and integrity to documents. This circumstance has made the digital signature a legally valid document.

The fact of being digital does not change anything and does not devalue the certificate, quite the contrary. Digital certification has multiple advantages in which is environmentally friendly and organizes everything in a simple and competent way.

2.2 Blockchain

Distributed networks, public and private key cryptography have been part of the society for years, however, the novelty of the blockchain is in its capacity to generate and communicate consensus on a common database updated through a decentralized network (M. Pilkington, 2015).

The main perk behind a decentralized network is that one of the parties engaged in a transaction only needs a simple verification with the history of the blockchain to approve it, without the need for an intermediary.

Unlike other systems, the registry generated by the blockchain is distributed, being preserved in millions of computers, as well as in data warehouses. There is no single owner of the records and each blockchain instance has a total of transactions of your market.

In a say, it is like a big ledger (where all accounting transactions are recorded), shared by all those who participate in the system, in which transactions are irreversibly recorded (M. Pilkington, 2015). It is the chronological record of all transactions compiled and validated that occurred in the network; as it is public, unique and shared by participants in the system.

The blockchain is like a reef of coral in which only the last records represent recent interactions, the previous ones are only a dead image of the past and accessed only on rare occasions to check historical data (M. Merz, 2016). A simple way of defining blockchains is comparing it to a system, with the objective to analyze and validate stored records.

The architecture of a blockchain is formed by these specifications (Pluralsight, 2017):

- **Proof-of-Work (PoW) protocol -** performed to calculate a difficult problem in the chain, to have the influence to add information to the network
- **Information verification and validation -** validity from date and time when it was posted to the network
- **Privacy of information -** practically impossible to construct related information within the network (although is easy to acknowledge such information knowing other people's identifications)

In addition, to better explain how it works, the Image 1, from Blockgeeks (A. Castro, 2017), represents an illustration of how the process of a blockchain is done.

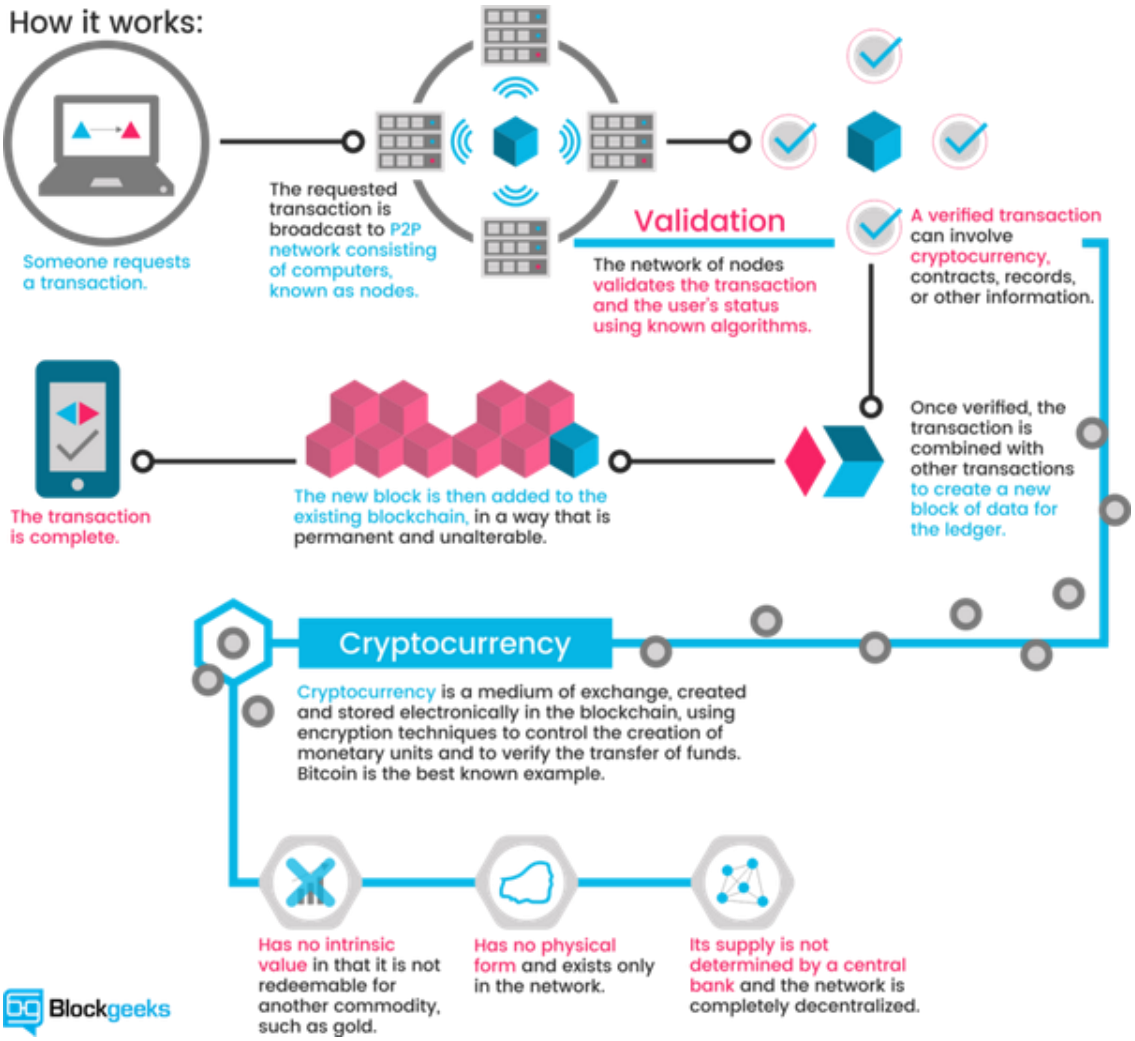


Image 1 - How the blockchain works

2.2.1 Distributed Consensus

Distributed Consensus is used in the distributed systems and is a critical aspect of the blockchain and cryptocurrencies. Consensus means that almost everyone agrees, however is different from unanimity since not all have to agree, and it is enough that the majority agrees.

In Blockchain, the consensus occurs between the participants of the Peer-to-Peer (P2P) network through methods composed of specific protocols and well-defined rules (M. Milutinovic, et al., 2016). All the P2P network nodes (Image 2) are involved in decision-making by consensus. It is the group or community function to decide the information that is approved.

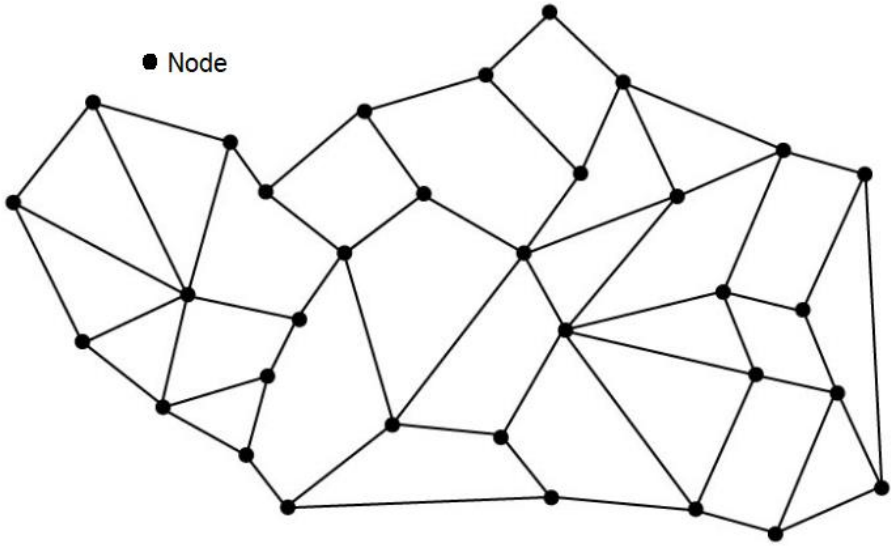


Image 2 - Example of a Distributed Network

From the point of view of those who develop applications on modern platforms, such as blockchain, consensus methods are a feature, service or configuration to be enabled and parameterized. They are often transparent (in programmatic terms) to the application of a developer.

2.2.2 Transaction

The data structure of a transaction reflects the semantics of the application. In the case of cryptocurrencies, this structure resembles as a credit balance sheet and is composed of the following elements: a timestamp, the hash of the previous transaction, the input value, the exit value, the destination address (which will receive the credit), and a signature of the private key of the account to debit the value (S. Meiklejohn, et al., 2013).

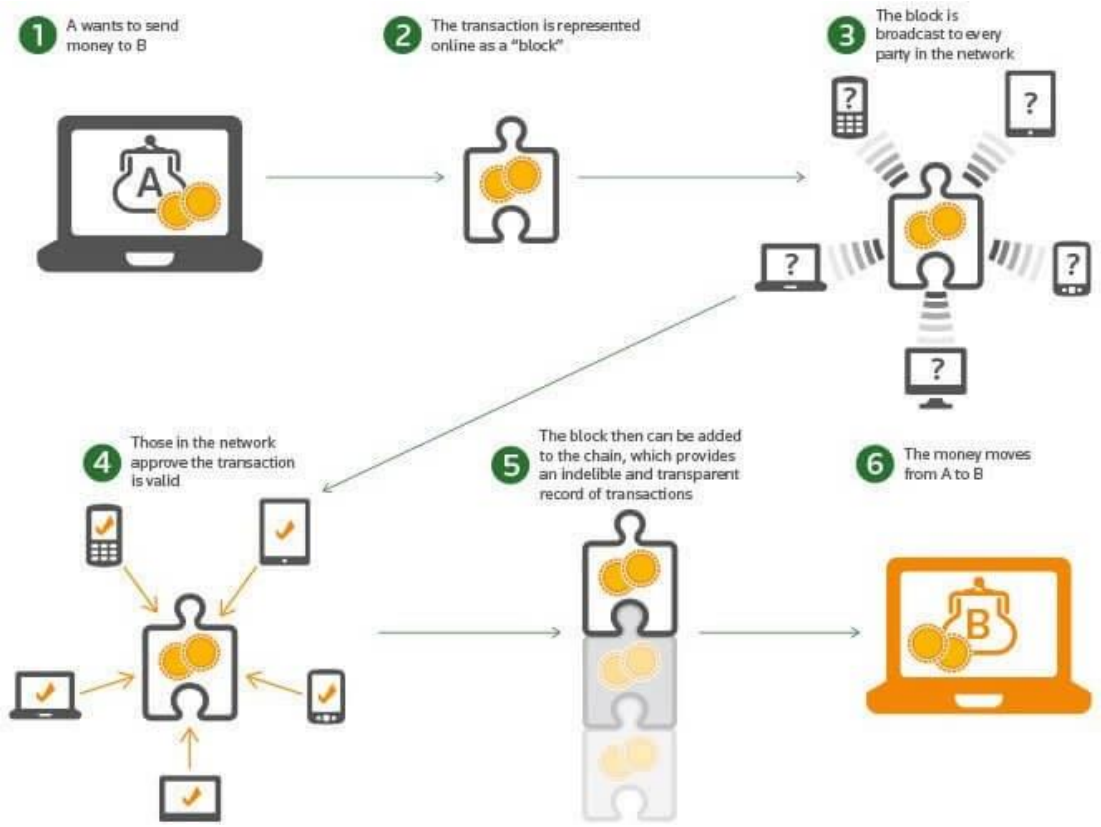


Image 3 – Blockchain's transaction Flow

Following the representation of Transaction flow between wallets in Image 3, from Honner (Alexandre & Andrew, 2016), the six steps are:

1. The specified amount stored in wallet A is prepared to be transferred to wallet B. The fee cost is calculated in this state that will be paid to the miner who validates it
2. The transaction is allocated in a queue, waiting to be added to some block
3. After the creation of this block, information is transmitted to all nodes that there is a block to be validated
4. The nodes enter a consensus to validate such block. This is the step in which the miners make the dispute of who validates the block first
5. After validation, the block joins the chain and is available in the ledger
6. Finally, the coin arrives in wallet B.

Due to the asynchronous nature of communication and the necessary time being relatively long, the consensus often ignores the confirmation steps.

Many frauds and other security issues in transactions could be avoided simply, by waiting for the necessary time and verification of the transactions (P. Franco, 2014), but this mostly composes a lot waiting time to the user, in which always needs to be avoided.

2.2.3 Chain of Blocks (Block Chain)

Transactions in a block are joined to each other in accordance with a binary tree structure based on hashes (known as Merkle Tree) (M. Scherer, 2017). The Image 4, from Medium (E. Kozliner, 2017), represents a demonstration of the Merkle Tree as well as the chain of blocks:

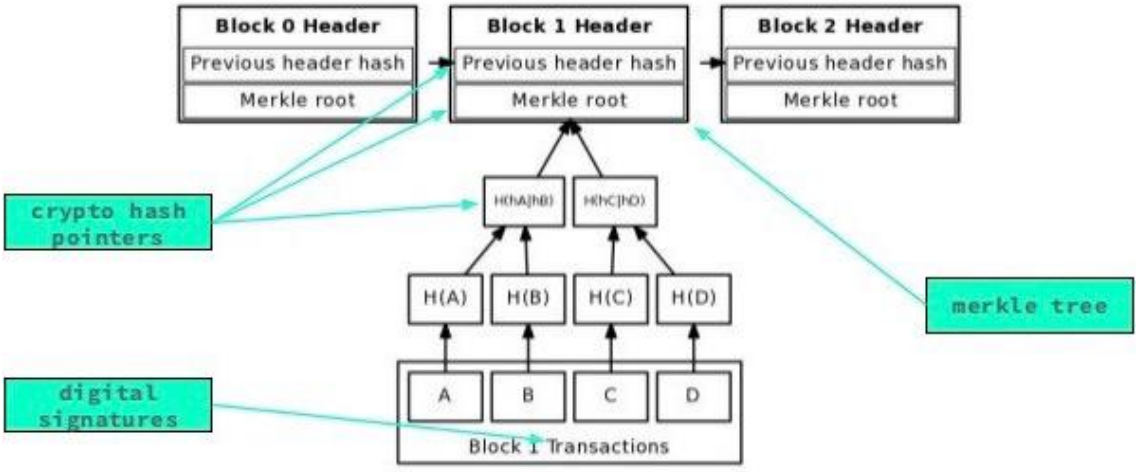


Image 4 - Chain of Blocks and Merkle Tree

The Merkle Tree's leaves are the transactions and the hashes of the parents are determined with the hashes of the children. For example, the hashes of the branches are calculated with the hashes of the leaves; the intermediate branch hashes are calculated with the hashes of the immediate branches; successively until the calculation of the hash of the root of the tree is included in the block.

The tree structure speeds up the verification operation if the transaction belongs to the block, which can be done in hash computations, where *n* is the size of the tree. Checking the hash of a transaction only uses the branch of the tree - Merkle Branch, as it is needed to verify the hash of the transaction to determine where the transaction is located.

2.2.4 Cryptography

Most blockchain solutions use two cryptographic routines: the cryptographic summary functions (commonly called hash functions) are used to generate addresses, this consist of calculated hash values from the public keys; and the digital signatures used to guarantee the authenticity and irrefutability of transactions.

Hash functions generate a sequence of bits, the value of the hash, which is unique to the function's input document, that is usually much smaller than the original document and has a fixed size (some hundreds) of bits. The hash function is unidirectional because it is not reversible, meaning that it cannot retrieve the original document from a hash sequence.

Asymmetric encryption (public key) for the digital signature is used to obtain integrity, authenticity, and irrefutability. A digital signature is the result of the certain cryptographic operation with the key is in clear text and the private key owner can generate messages, which can be verified by anyone who knows the corresponding public key.

The user cannot deny the authorship, as there is a digital signature made with your private key, for this reason, the signature is irrefutable, and the signature can be verified by anyone with the public key (A. Kosba, et al., 2016).

2.2.5 Mining

Mining is the process responsible for validation and updating the blockchain whereby some nodes are called miners. Their function is to validate transactions and generate a new block to include in the blockchain, until a viable block is obtained to be sent to all nodes in the network (M. Pilkington, 2015).

A lot of energy is performed, in the form of PoW, for this reason, miners are rewarded (in Ether - Ethereum blockchain) to the valid and full integration of the block. The miners, upon receiving a message with a transaction, store them in a transaction database, that has not yet mined. Transactions remain temporarily, in a sort of priority queue, until they are withdrawn to be included in a new block.

Each miner has a different queue of transactions and can select which transactions will include in this new block. After selecting which transactions will be included it will generate a Merkle Tree and include the value of its root in the header (Image 5, from (Blockstars, 2015)).

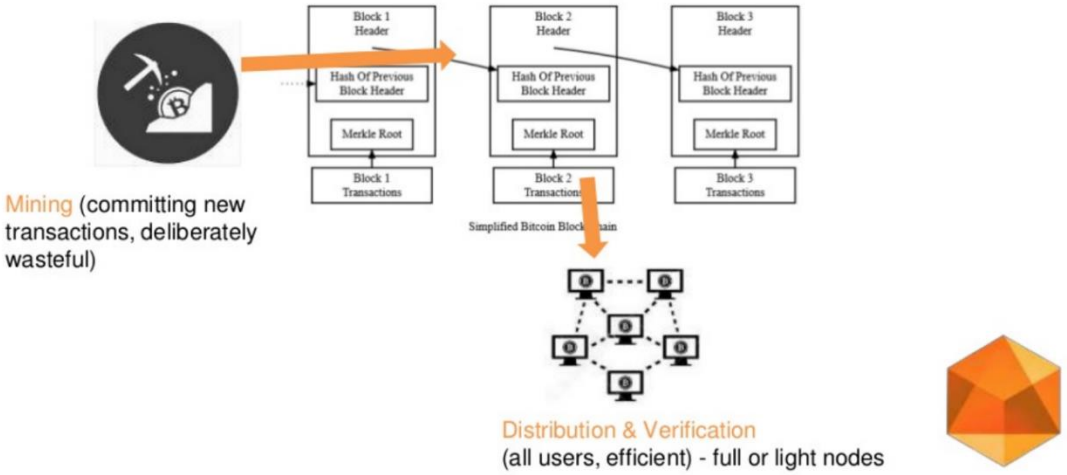


Image 5 - Block to the Blockchain (Mining validation)

2.2.6 Versions

The blockchain is a target for constant adaptation and its core is being refined every version to improve and adjust new applications. Versions in which have proven to have a significate impact on the evolution of the blockchain technology (M. Swan, 2015) (Unibright.io, 2017):

- **Blockchain 1.0: Currency**
    - The 1st generation of the blockchain protocol, created in 2009, and related to the Bitcoin protocol
    - Implementation of DTL that enables the use of currency transaction (cryptocurrencies) among individuals
- **Blockchain 2.0: Smart Contracts**
    - New applications of the blockchain protocol outside the monetary transactions and the financial system
    - Applies in the Ethereum protocol, created in 2013, that brings the usage of smart contracts within the blockchain
- **Blockchain 3.0: DApps**
    - Implemented in 2017, constitutes the use of decentralized applications to store and communicate between entities, serving as a frontend application to users, with backend calls to several procedures
    - Improves the blockchain protocol for a faster consensus and quicker transaction confirmation
- **Blockchain 4.0: Industry Usage**
    - A 4th generation that is in testing/development, to improve the blockchain network into finding a place in the industry areas
    - Promises to bring all the required functions to create a fully automated and integrated system in the blockchain network for the industrial demands
    - When completely distributed, this version is going to revolutionize the blockchain protocol, in the appearance of new areas of application

Since the beginning, different kinds of existing and potential activities in the blockchain are emerging, the cause of this is the improvement of the blockchain protocol that keeps evolving in each version.

2.3 Blockchain application in education

The first situations reminded in blockchain's applications are those referring to the financial sector, followed by the registration of documents and the tracing of products in supply chains. This is mainly because these are the best situations explored so far and have already given a rise to startups and pilot projects by large companies.

However, education can have some benefit in using this technology (A. Grech, et al., 2017). Some of those possible uses of the blockchain in education are: consulting certificates or qualifications, handling student records, managing intellectual property and payments.

Even if it is something extremely new, the use of blockchain in education will most likely promote a disruption in the student information. Among the changes:

- Induce the end of certificates issued on paper
- The possibility of validating a certificate in the blockchain, avoiding increasing fraud related to degrees
- Automatic tracking of citations of articles and/or teaching materials
- Identification of the origin of certain knowledge (intellectual property) and of the applicable copyright
- Reduced costs of storage and management of educational data
- Use of cryptocurrency to finance studies
- The secure sharing of knowledge between partner institutions, following the evolution of shared processes over time

These changes promote a series of advantages. A few examples of advantages, in detail:

- The blockchain is a good way to store information. Records can be stored forever since each computer in the blockchain network has a copy of the entire network updated. In this way, educational institutions can keep records of grades, credits and other information about their students
- For educational institutions, it is a good way to save diplomas and documents and make it accessible to employers. Blockchain allows you to keep thesis and other academic papers and prove to whom they belong. Plagiarism and conflicts over intellectual property is no longer a problem
- Save enough money on servers and keep information secure, as each Blockchain participant owns their own information, this will enable schools and universities to
- Usage of blockchain is best known for ease of transaction in payments. This will allow scholarships and company funding for investigations to be done without intermediaries, quickly and cheaply as possible

The best-known case is to register diplomas in the blockchain, this allows a candidate to share their certificates, for a potential contractor to verify the certificates directly on the network and makes the process of forging a certificate more difficult than it is. One of the most advanced solutions for registering diplomas and certificates is the Blockcerts open standard (MIT Media Lab Learning Initiative, 2016).

A blockchain allows the grade obtained in each assessment of a subject to be recorded. The student may have a copy of his entire academic life in his or her pocket, which would eliminate bureaucracy when requesting a duplicate of the school record and facilitate the process of transferring from one institution to another.

The possibility of having diplomas or certificates issued in blockchain is extremely interesting for both workers and employers (G. Chen, et al., 2018). This opens to numerous possibilities in terms of selecting candidates for a job vacancy or for offering customized courses. This will probably create a shift in education and many business opportunities, that will arise for those who invest in this market, whether as a student or as an entrepreneur.

2.3.1 Certification Purposes

The growth in the infrastructure of the networks has brought an increasingly large obstacle in reliable information. Not having the possession of the physical paper promotes lack of trust to the contracting companies in ensuring the credibility of the issued certification; this is the main problem that involves the certification stored in the web.

However, this never imposes an obstacle to the new technologies, such as projects in the Ethereum blockchain that have made possible to work and reuse an existing decentralized infrastructure for other challenges, making it even more resilient to distribute reliable information (D. Tapscott & A. Tapscott, 2017 [b]). Also, it is proven that people are becoming more familiarized with the technology, giving all the consideration done to the blockchain in ensuring the validation of such documents in the network.

Creating a certificate is relatively simple, everything can be based on the creation of a digital file that contains basic information that is signed through a private key that only the sender has access. Only through mechanisms based on encryption, it is possible to validate who is the sender and whether your content has been tampered with.

Using a decentralized blockchain has the ability to digitally distribute certificates reliably through fraud-validation premises, efficient public access, to divide ownership of the certificate between issuer and certificate is the way for reputation to be distributed.

Certification in blockchain tackles problems addressing the person’s access to records, loss of certification history, slow and difficult delivery of transcripts (C. Jagers, 2016). These problems represent the convenience and the necessity that the individuals need to have control of their own records and those records don’t have to be owned by any company or government.

2.3.2 Blockcerts

Developed project within the blockchain technology that addresses the certification' context - Blockcerts (MIT Institution, 2016), developed by MIT Learning Lab and Media Lab.

Fully implemented in the Bitcoin blockchain, in 2016, and has plans to soon be expanded to Ethereum. This project enables the management of certificates within the blockchain for academic credentials, professional certifications, workforce development, and civic records.

The project was released with the ability to be used by any school and allows the student to maintain and share their own official records directly with others (MIT Media Lab Learning Initiative, 2016). This technology creates a new trust infrastructure that replaces the need to request the records from a central authority. These digital records are registered in a cryptographically signed, tamper-proof and shareable block, this means that the records remain authenticated, unmodified and public.

The principal objective is to enable the continuous innovation that offers individuals the ability to own and share their own records. Also support the goal of this community to create technical resources that other developers can use in their own projects, rather than independently developing custom implementations.

Blockerts works by creating a digital file the contains some basic information (name of the recipient, name of the issuer, issued date) and sign the contents of the certificate to the blockchain. The system keeps track of who issued and received the certificate and validates the content of the certificate. The Image 6, from Blockcerts web page, shows the flow of how the project works.

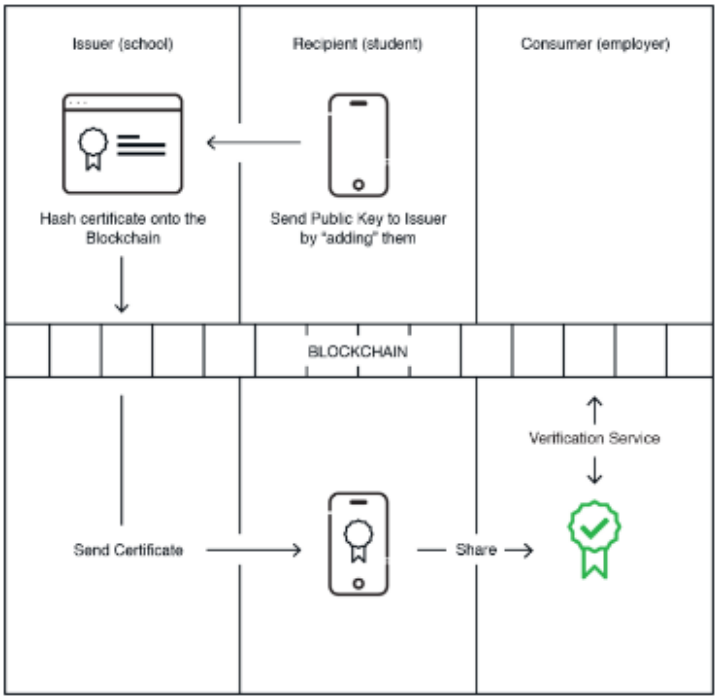


Image 6 - Blockcert Process Flow

2.3.3 TrueRec

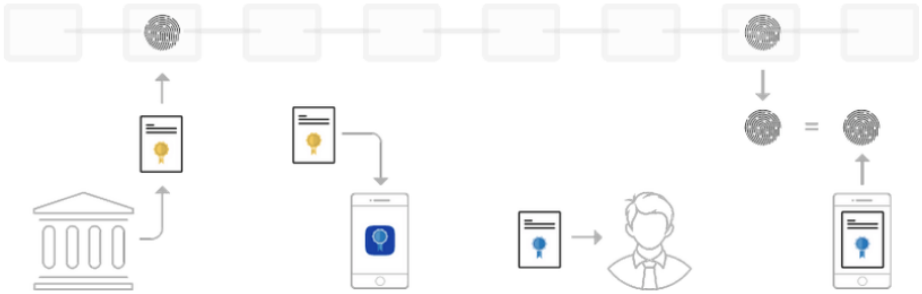
Developed by enterprise SAP and focused entirely in the education areas, TrueRec is an implemented project that allows issuing academic achievements to the Ethereum public blockchain. Allows users to keep their respective academic records, issued by academic institutions, in the TrueRec application on their device.

Using blockchain technology and Cloud Foundry, this completely developed solution (in 2017) offers Trusted Digital Credential ensuring confidence, authenticity and integrity, and in conjunction with the TrueRec application, this complements in having convenience, privacy and trust to the credentials of the user (C. Guiterrez & A. Khizhniak, 2017).

This implementation's process is structured in the issue of the credentials by the certified institutions and the acceptance of the issued credential by the learner, to their account. This implements a simple and safer validation method, where the credential is only issued by trusted institutions and if the credential is not accepted by the learner the certification is not integrated.

The application inside the learner's device has all the accepted certificated, stored locally in their device, proving an effective way to manage and verify the credentials that can be displayed or shared to other employment organizations, via a link.

Image 7, from the TrueRec web page, represents the process flow of the TrueRec project.



1	2	3	4
When your institution issues your credential through TrueRec, a digital "fingerprint" (or hash) is added to the blockchain.	We send you the credential as a digital file that you can import to the TrueRec app. Your credential is safe because it's stored only on your device.	You can securely share your credentials from the TrueRec app. You can share all your credentials or only certain credentials.	Recipients can verify the validity of your credentials by comparing the document against its fingerprint on the blockchain.

Image 7 - TrueRec Process Flow

2.4 Ethereum

A platform dedicated to the development and deployment of decentralized applications of high reliability – Ethereum, uses a blockchain to execute contracts inviolably since all nodes in the network must agree on the results of each computation.

Anyone can develop a blockchain-based system and run it using Ethereum's own environment without having to deal with the costs of developing their own. There are already several systems that use Bitcoin in this way, but they are limited by the reduced capabilities of Bitcoin scripts (without serious scalability problems and loss of the innate security to a blockchain) (Ethereum Comunity, 2014).

The system implements a token called Ether, which is used both to compensate the nodes for computations performed and to perform the values' transfer between accounts/wallets.

Applications in Ethereum blockchain work exactly as scheduled without any possibility of downtime, censorship, fraud or interference by third parties. They can even be customized, allowing developers to create markets, store records of debts or promises; make transactions according to certain instructions and other things. All of this without an intermediary or counterparty risk (G. Wood, 2014).

Some of the possible applications the Ethereum can offer are illustrated in Image 8, from (D. Palmer, 2016), with the Ethereum' icon in the middle. A few examples of potential applications are financial transfers, historical records, and crowdfunding campaigns.

Image 8 - Few applications in Ethereum

2.4.1 Ether

Ethereum defined an internal price mechanism called Ether, to avoid denial of service attacks and other types of spam. In a simplified way, it represents the price that is being willed to pay for each computation.

A simple computational step should cost some Ether (at least 1 Ether), while more computationally complex operations should cost higher values (Ethereum Community, 2016). There is also a 5 Ether charge for each byte in the transaction data.

The basic part of the initial verification is checking whether the applicant has the required amount of money to subtract it from the account and pay transaction fees, as Ether cost.

Whenever some piece of code is next to be executed, the part that is requesting such execution shall establish the maximum number of Ether units that are willing to use and what Ether value it is paid for each unit spent.

2.4.2 Smart Contracts

Computer programs (written in general or specific programming languages) that can be correctly executed by a network, without an external entity to arbitrate the agreement, and addresses issues that require agreements with minimal trust between the parties involved in a distributed system. These programs may correspond to the contract itself, meaning that people enter into an agreement is embodied in computer codes, in respect of these specific clauses.

Smart contracts allow automatic code execution in the blockchain, usually for causing an external action during a defined firing event. These may be external calls to software applications that make additional the execution of other smart contracts. The power of the program code of a smart contract can vary considerably, for example Ethereum money is consumed (Ether) when executing smart contracts.

In traditional server architectures, each application must configure its own servers to run their own code in isolation, which makes it difficult to share data and, if a single application is compromised or goes offline, many users and other applications will be affected (F. Tschorsch & B. Scheuermann, 2016).

To simple demonstrate how the smart contracts perform, the Image 9, from Blockgeeks (A. Rosic, 2017 [b]), shows how exchange certificates procedures are done in smart contracts.

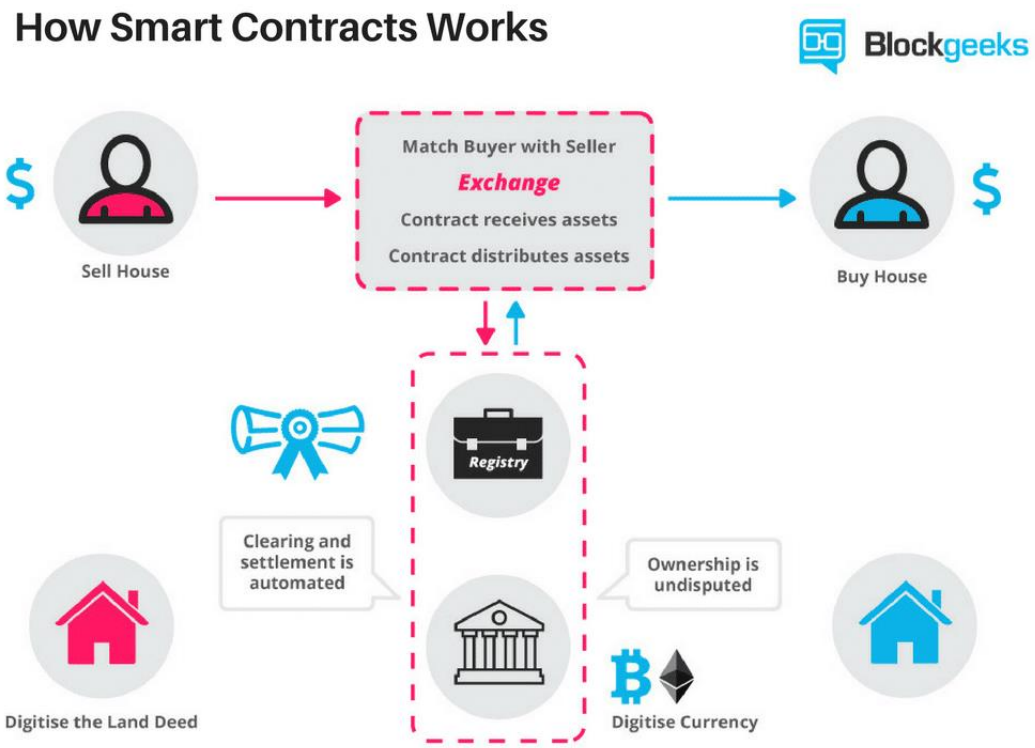


Image 9 - How Smart Contracts works

The facts that make smart contracts actually "smart" are the advantages that it offers to several tasks (E. Mik, 2017), a few are:

- **Speed and real-time updates -** smart contracts use software code to automate tasks, can increase the speed of a wide variety of business processes
- **Accuracy -** automated transactions are not only faster but less prone to manual errors
- **Less risk of execution -** the decentralized execution process virtually eliminates the risk of tampering, lack of performance or errors, since execution is automatically managed by the network rather than an individual party
- **Fewer intermediaries -** smart contracts can reduce or eliminate reliance on external intermediaries who provide trust services as collateral between counterparties
- **Lower cost processes -** less human intervention and less intermediary, and therefore reduce costs. In addition, the amount of time it takes from the start of the contract to completion is also reduced, saving money and reducing associated back-office costs
- **New business models -** provide an economical way to ensure that transactions are performed reliably as agreed, they will allow for new types of business. Insurance can be done P2P, rather than through a centralized institution

The smart contracts aren't kept by the community, instead, they are encrypted and sent to other computers via a network, the blockchain. Every new information that is brought into the blockchain, as a new smart contract must be agreed between two parties, so then it can establish a new block. This will bind to the remaining building blocks of the blockchain.

Image 10, from Blockgeeks (A. Rosic, 2017 [a]), enforces how the smart contracts process on the blockchain, in a technical view.

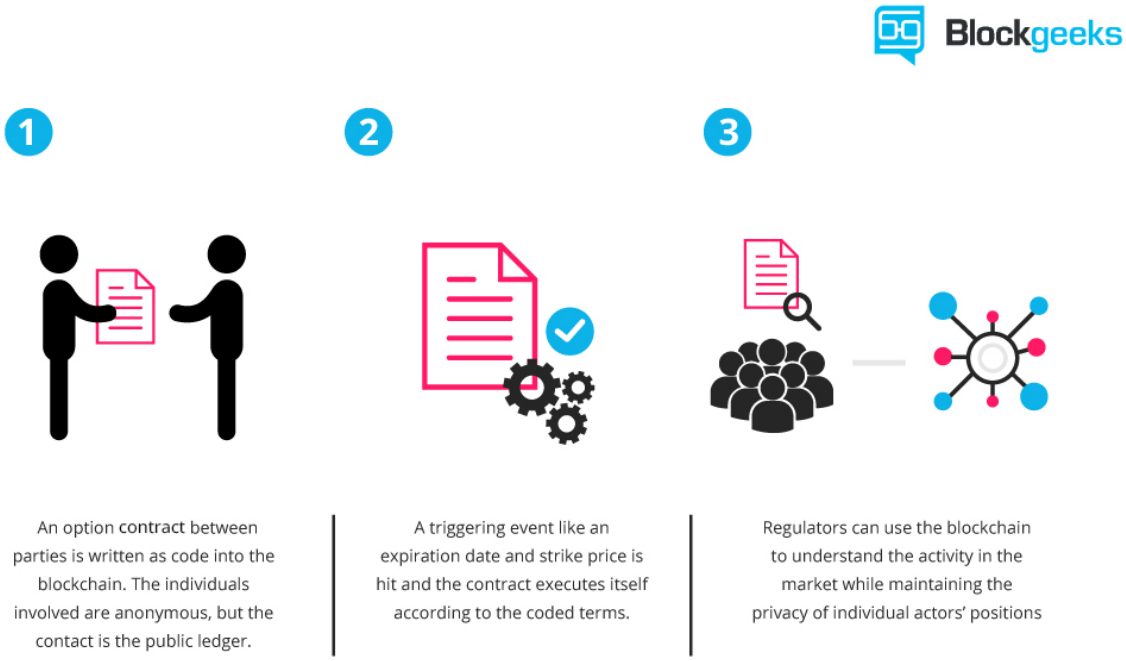


Image 10 - Blockchain's Smart Contract process

The superstition that smart contracts correspond itself to a contract, is mainly because the parties enter into an agreement. This kind of contract is embodied in computer codes, to simply refer to one or more clauses written, that is enforced, with respect to some specific clauses.

Ethereum created its own programming language called Solidity (Solidity/Ethereum Community, 2016). It is with this language that smart contracts are written and placed in the Ethereum network and for them to "run" they are executed in the Ethereum Virtual Machine (EVM).

Although smart contracts have not been used on a large scale, the potentialities and the investment gathered around them justify the questions concerning its framework.

2.4.3 Decentralized Application (DApp)

Ethereum builds DApp within its blockchain, these are an open source autonomous entity that can be updated or modified by a group of users through consensus.

The purpose of a DApp is to remove the need for central authority, such as governments or corporations. As transactions between people as the medium of the application without an existence of an intermediary (Ethereum Comunity, 2014).

The common features of the DApps are (S. Velu, 2017):

- **Open Source -** disposes of a source code that is open to all users to change (in a consensus matter)
- **Decentralized -** records within the application are stored on a public and decentralized blockchain, avoiding centralization drawbacks
- **Incentivize -** validators of the blockchain (miners) are incentivized in rewarding them accordingly with cryptocurrency tokens
- **Protocol -** application community agree on proof of value protocol of the blockchain (mostly Proof-of-Work (PoW) protocol [6])

2.4.4 About Ethereum History

Ethereum was first described by Buterin Vitalik (R. Hackett, 2016) at the end of 2013 because of his research and the work in the Bitcoin community. Shortly thereafter, Vitalik published the Ethereum white paper, in which he describes in detail the technical and rational design behind the Ethereum protocol, in addition to the two smart contracts.

Vitalik officially presented his idea in 2014. Immediately after their presentation, a large of developers seeking to better understand the proposal.

In the same year, Dr. Gavin Wood, who began working with Buterin, published the Ethereum yellow paper (G. Wood, 2014) which served as the technical bible and specification for the EVM. From this technical article, Ethereum has the possibility of being implemented in several programming languages.

In addition to developing software for Ethereum, the feasibility of launching a blockchain and a new cryptocurrency required a gigantic effort, a kind of application initialization mechanism, to gather the resources needed to put the platform upright and in operation. To initiate this cause, a large network of developers, miners, investors and other interested parties, allowed that the Ethereum to have an announced plan to conduct a preview of the Ether coins.

[6] A representation of the Ethereum's blockchain structure, with the PoW protocol, can be found the Annex B - Ethereum Yellow Paper Representation.

2.4.5 Roadmaps/Milestones

Ethereum is represented by five roadmaps, each representing different milestones in phases for the Ethereum's personal project (J. Ray, 2018):

1. Prerelease: **Olympic**, May 2015 (V. Buterin, 2015)
    - The original testnet of the Ethereum network, allowing developers to test the project's limits. Concluded this phase offered a lot of improvements to the network
2. Release 1: **Frontier**, July 2015 (Ethereum Frontier Release, 2015)
    - First live release of the Ethereum network (in a beta stage). Developers could experiment, with writing smart contracts and decentralized applications, and miners began joining the network Ethereum to maintain the security of the blockchain network, earning ether from the blocks mined.
3. Release 2: **Homestead**, March 2016 (V. Trón & H. Jameson, 2016)
    - First production release of Ethereum (leaving the beta stage). The robust version with a series of protocol improvements, granting the network a lot of upgrades for speeding up transactions.
4. Release 3: **Metropolis**, October 2017 (Blockgeeks, 2017)
    - The third greater release of Ethereum with a lighter, faster and more secure network. It is composed of two sub-releases: Byzantium (October 2017) and Constantinople (planned on October 2018). Constantinople intention is to have a higher level of scalability, increase efficiency and lower transaction fees (lower rewards for mining), and will serve as a smooth transition from a PoW to Proof-of-Stake (PoS), for the later Casper release
5. Release 4: **Serenity**, to be announced (V. Buterin, 2016)
    - An intentional future release to bring the Ethereum network a complete shift from the PoW consensus algorithm to the PoS algorithm, utilizing the Casper previous release in the hybrid PoW/PoS mechanism.

3 Value Analysis

The value analysis overview is explained in this chapter, so that the reader can obtain an insight into the project's liability in producing value.

3.1 Business and Innovation Process

To define the business innovation process, Peter Koen New Concept Development (NCD) model (P. A. Koen, et al., 2002) fundaments an understandable analysis of the development stage in a project. By this matter, is taken 5 key elements of this model, to which each one will be explained, in the methods taken.

- **Opportunity Analysis**

  An opportunity is evaluated to confirm whether it is worth following it up. Additional information is required to translate opportunity into business and technological opportunities, which can be acquired through focus groups, market studies (for example: behavioral consumers, market trends) or scientific experimentation.

  The method to ensure this element is the subchapter Current Market Value.

- **Opportunity Identification**

  This element is typically driven by business objectives, and it is through it that the organization identifies the opportunities that must be captured. In many cases, this element precedes the generation and enrichment of ideas.

  The method to ensure this element is the subchapter Quality Function Deployment.

- **Generating and enriching ideas**

  This element is related to origination, development and maturation of a concrete idea. The generation of the idea is evolutionary and iterative and can be a target of mutations (combinations, modifications or updates) as it is examined, studied, discussed and developed along with other elements of the model.

  The method to ensure this element is the subchapter Canvas Business Model.

- **Idea Selection**

  The difficulty for most organizations is the selection of ideas that are foreseen to have a greater business value. A good selection is critical to the health and success of the organization.

  The problem is that there are no processes that ensure a good selection. Mostly the idea selection methods involve iterative series of activities that include several passages through the previous NCD elements.

  The selection process may be as simple as choosing one among the generated ideas, or as complex as a multi-stage business process. However, due to the scarcity of information and low level of understanding that characterize the beginning of product development, decision processes are difficult to implement.

  The method to ensure this element is the subchapter Value Proposition and Customer Value.

- **Concept Definition**

  The final element of the development NCD and is the previous step in the New Product Development (NPD - process delivering the product/service to the market). Concept Definition, in short, represents the specification of the concept (product concept, concept specifications and architecture) resulting from the best ideas and a business case based on potential market and competition assessments, investment requirements and risk analysis of the project.

  To this project theme, Concept Definition uses the analysis behind each of the previous NCD element's methods and takes the benefit in analyzing the Game Theory subchapter. This general thought, each method makes the concept definition a complete reflection, for every analysis taken in each following subchapter.

More information about each method shall be explained in the following subchapters.

3.2 Current Market Value

As mentioned before, Ethereum application is currently having a progressive increase in the people's way of understanding and use the blockchain's transactions. By this affirmation, DApps are proven to have new fields of application, open to be developed by other programmers.

The cryptocurrencies likewise are proven to have an impactful claim in the way people trade values between them. Every day more and more cryptocurrencies are being made, mainly as an Initial Coin Offering (ICO), and the value most of them is greatly increasing as the time moves forward (C. Bovaird, 2017). Table 1 shows the Return on Investment (ROI) based on a few cryptocurrencies in a recent study (A. Lielacher, 2017).

Table 1 - Top 5 Biggest ICOs

ICO	ROI (%)
NXT	1 477 000
IOTA	332 500
ETHEREUM	152 500
NEO (ANTSHARES)	114 000
STRATIS	81 000

Even though ICOs are a risk funding, it has confirmed to have more return value than the loss in most cases (as shown in the table reflects an extreme return value).

In contrast, blockchains are a recent application, unknown to many how it can be impactful for the civilization. Nevertheless, all the fields that support it make so that the blockchain never goes underused, as it shall stay increasing and refining its core daily.

By the previous statement, every DApp developed within the blockchain has a lot of benefits and support by the community and continues to evolve as time goes on. Even if not found records of market value regarding the exchange of contracts within the blockchain, the values alone in the cryptocurrencies compensate for the usage of the blockchain technology to develop an application in the educational fields.

3.3 Quality Function Deployment

The Quality Function Deployment (QFD) model (Warwick Manufacturing Group, 2007) has the importance to design the quality of the service and to show in detail where the service development process. Image 11 has the representation of the service requirements in this project to the relationship in the customer requirements.

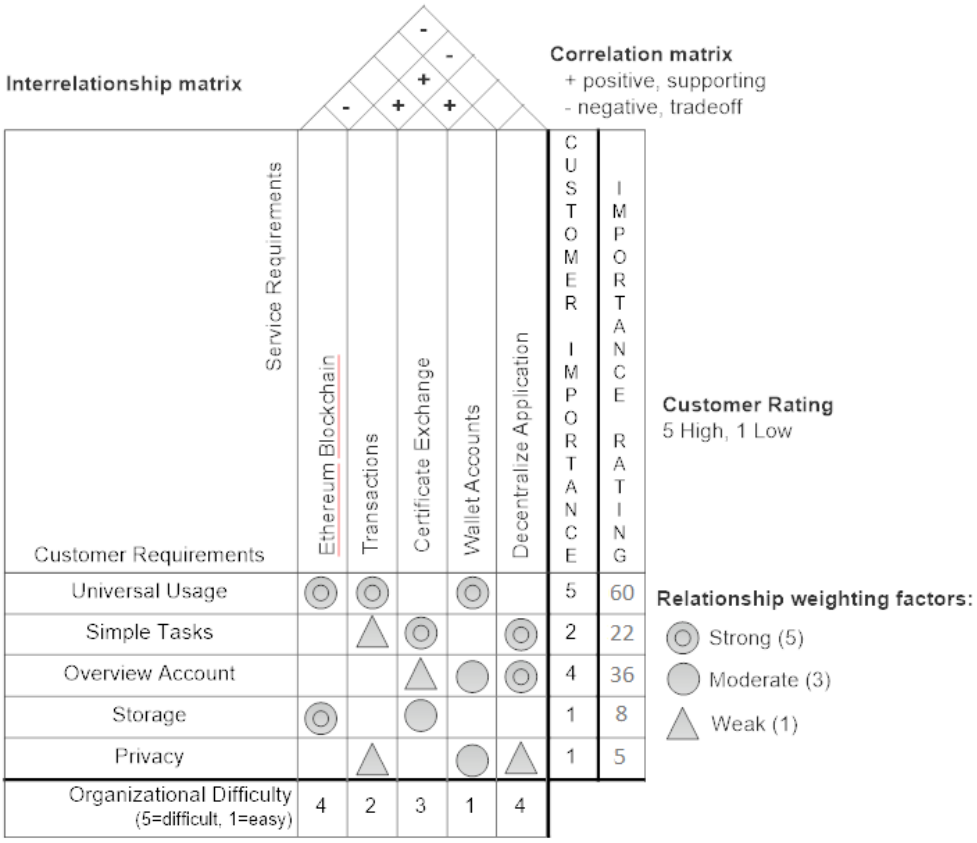


Image 11 - Project's QFD model

In the point of view of Image 11, the shown Services Requirements and the Customer Requirements are displayed with their Relationship Weighting factors, which attains how strong the Service Requirements references the Customer Requirements.

The Organizational Difficulty displays the difficulty in performing/developing the Services Requirements, for the prototype standpoint.

Customer Importance displays the features that customer is most prone to want; and Importance Rating shows the calculation of the rating done to each customer requirement, through the sum of symbols in each row multiplying by Customer Importance.

According to the Customer Importance, from the customer perspective, the most valuable requirement is the Universal Usage and it has the most Importance Rating since it has some fulfillments done by the service requirements (including the higher value of 60 points). This makes so that the service mostly follows the behaviors of the customer needs.

3.4 Canvas Business Model

The Image 12 represents the Canvas Business Model to help discriminate the different factors that compose this project, in the service value proposition as well as the business strategic management.

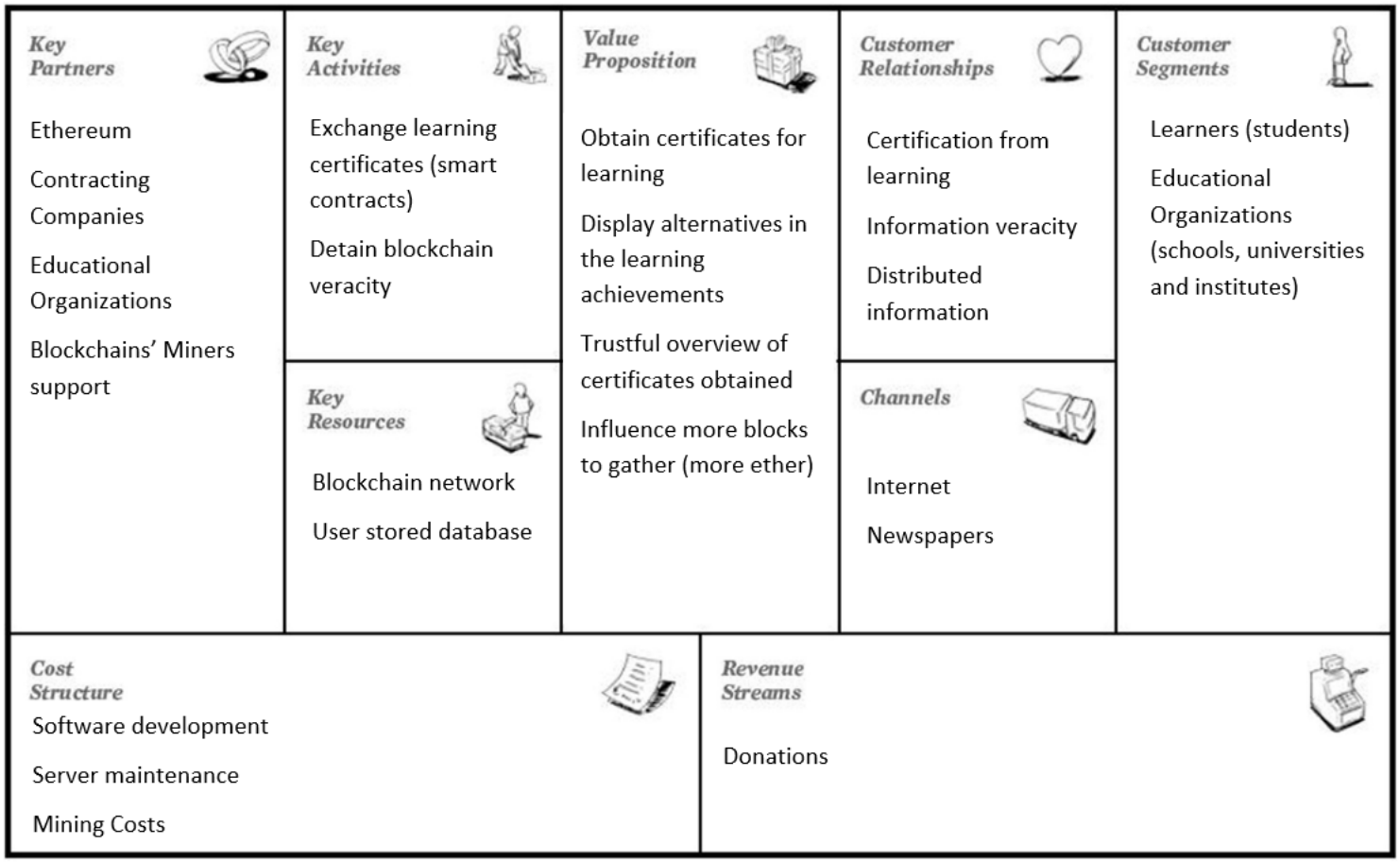


Image 12 - Canvas Business model

3.5 Customer Value

The customer value definition is related to the necessary desirable characteristics, that the product/service offer, in terms of main of attributes. Attractive attributes can influence the customer acquiring the product/service, while its counterpart contributes that the customers avoid the thought to obtain certain purchase (M. Dejen & H. Sekandary, 2008).

Still, the main attributes of a product are not the huge problem to identify; the problem is identifying the needs of every customer, as every customer is not treated with the same personality. The attributes must target each of the segment, that represents a group of customers of similar traits. This alone requires a lot of effort in identifying the needs of each segment of customers to obtain the most value.

In the project's thought, the most focused segment is the clearly the learner (a person who obtains the learning certificate). The next customer segment is the educational institutions (entities who send the certificates for the learners). From this standpoint, the customer value must be divided into two segments.

The outcome for the first segment – Learner Segment – is the to obtain the full detail of the certificates; and the other one – Educational Institution Segment – is to send those certificates to the Learner. For the developed prototype service, these outcomes are taken into consideration.

To determine the main benefits (advantages) and sacrifices (disadvantages) of the developed prototype, an analysis between attributes that the customer requires is taken in reflection. As for the attributes, both customer's segment follows the same attributes:

- **Reliability** – trust ensured to the service processes
- **Security** – conviction to the services safekeeping of the user's content
- **Effective** – effective to the actions performed in the service
- **Responsiveness** – capability of support in the origin of an error
- **Price** – purchase or maintenance of the service (low is preferred)

In the service' prototype has the benefits of Reliability, Security and Price. These benefits are mainly offered by the blockchain application (as explained the benefits in the above chapter).

As for sacrifices attributes it has the Effective and Responsiveness. Effective is not going to be ensured, since the process is rather complicated within the block records, and Responsiveness, after the project's prototype is completed, if not contributed by any investors, the maintenance to this project is going to be compromised.

3.6 Value Proposition

The value proposition is understood as the set of products and services that create the service value, representing the benefits delivered by the company from a direct relationship in the way the products and/or services are offered, to meet the customers' needs (A. Osterwalder, et al., 2017).

Taking this to account, the project delivers the main benefit in the approaches displayed in the Image 13[7].

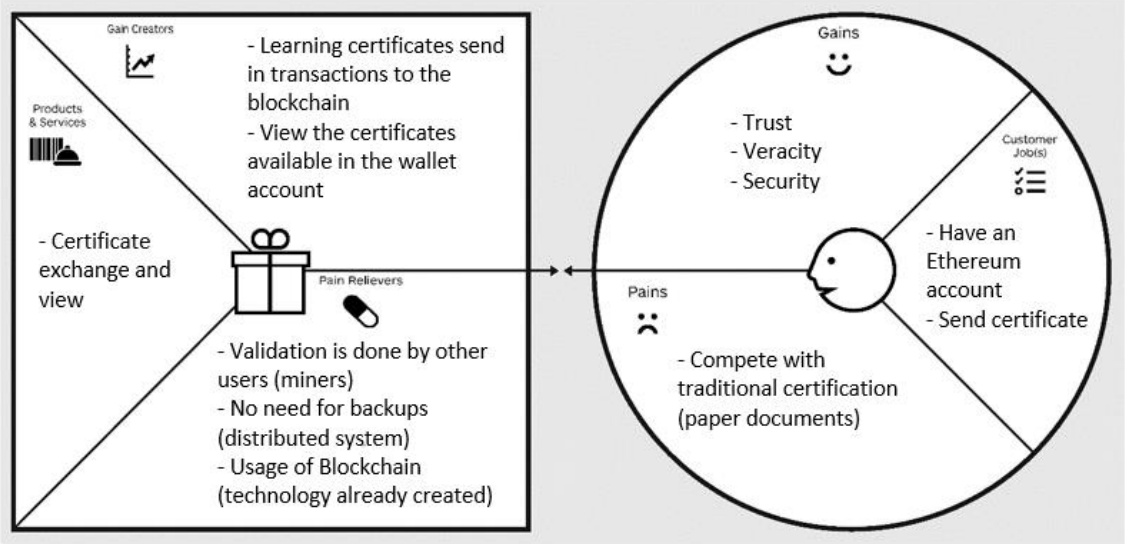


Image 13 - Value Proposition - Service Canvas

[7] In the Image 13 the "present" illustration shows the service practices, as for the "face" illustration displays the Customer perspective. All the characteristics shown are related to the value proposition.

3.7 Game Theory

The correlation between users in the management of certificates distributed in the blockchain differs between the opinion of everyone. Since everyone is likely to have an unspecified opinion, a randomized result can be added to better acknowledge the probability to determine the better likelihood to support/oppose the idea.

Game Theory (T. L. Turocy & B. Stengel, 2001) comes into play to provide the analysis behind this problem.

Fact on the project's perspective, the Learner[8] acknowledgment of certifications stored in the blockchain is based on the Educational Institution[9] that sends the certificate and Contracting Organization[10] that use the application to overview the certification.

The Learner is more likely to obtain and store their personal certifications in the blockchain when all the Educational Institutions uses this kind of system to send them. On the other hand, if the Educational Institution doesn't approve the send of certificated in such, the Learner is not going to use this application. Also, the same opinion goes to the Companies that are likely to influence the idea that Learners need or don't need to use their accounts to show them the proven certifications/diplomas.

Nash equilibrium better includes in this theory, since the Nash equilibrium is relatively evident in one random variable – "Support" or "No Support" (Oppose). The support will always be associated between the user's choice, and different scenarios are shown to demonstrate the choice influenced by the Learner in the Contracting Organization and Educational Institution opinion.

[8] Individual who uses the application to view certificates.
[9] Certified educational institutions that send the certification to the blockchain.
[10] Organizations that are employing people.

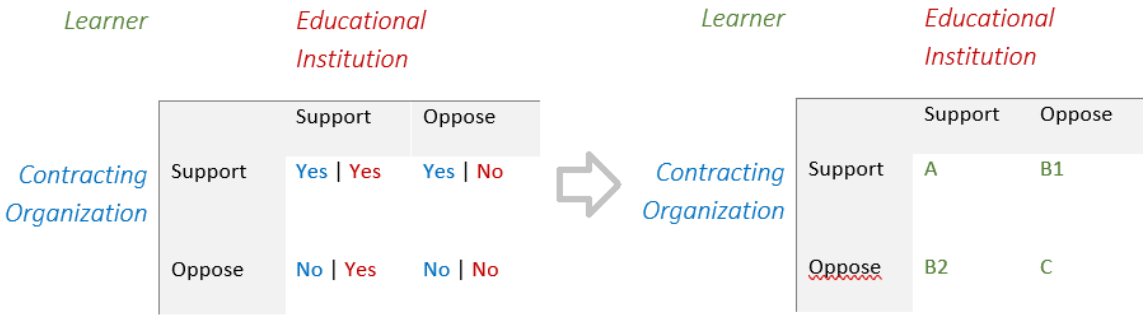


Image 14 - Game Theory project's dilemma

Following the representation in the results on Image 14, the four scenarios influence the Learner in certain ways:

- **Scenario A -** offers full support from both Educational Institution and Contracting Organization, meaning that the Learners choice has a complete influence in using the service
- **Scenario B1 and B2 -** the Learner's choice is influenced partially by the other entities' choice, as some might use or not
- **Scenario C -** Learner doesn't need to have any use of the service, meaning that it is a disposable application with no personal use in their life

To the analysis of these Scenarios, if the prototype ideally establishes the support of the Educational Institution and the Contracting Organization, the Learners will surely use the prototype to store their certificate; as for its contra-part, affects the Learners choice in not using the application at all.

So, by this affirmation, the prototype' service must demonstrate enough support from both entities so that the Learner continuously uses service.

4 Requirements Analysis

This chapter explains the developed prototype's requirements, with an overview in the project's implementation intentions and testing. Contains a design to the solution that complements a better development comprehension. The following sections explain the development purpose and structure, the possible alternatives and development comparison between similar projects.

4.1 Prototype Design

The storage of certificates blockchain, in a digital image format that can then be viewed later, is the principal approach of this prototype's development. To accomplish this, it desirable to have a designed control service that serves as an intermediate interface for the insertion and to obtain the certificates information stored inside the blockchain.

The control service is accessible by the user's browser and its functions are to create, modify, inactive[11] and view certificates inserted in the blockchain.

Certificate image insertion can only be done by the educational institutions, as the certificates are always provided by the institutions. Therefore, the designed service has an authentication access to ensure that the insertions are done only by the permitted educational institutions, that no certificates insertion from fake institutions are formed within the application. Also, the institutions can modify the information, as long its later validated to the blockchain.

As for the certificate view, the certificate's images stored in the blockchain are viewed in any browser, as long there is provided an Ethereum account hash to the control service of the desirable account to view. The certificate's displayed information, in the browser, can be used to

[11] The blocks and its transactions are immutable, so the inactive state serves as a replacement to delete the contracts, with the meaning they are not used. However, the contracts can still be reactivated if necessary.

view certificates from a student's wallet, thus proven easier for employers to interpret the available certificates.

The general intended idea can be interpreted by the high-level vision, illustrated in Image 15. The Control Service acts as an interface between the users (Educational Institution and Learner) and the Blockchain network, in order to obtain the necessary information regarding the certificate smart contracts.

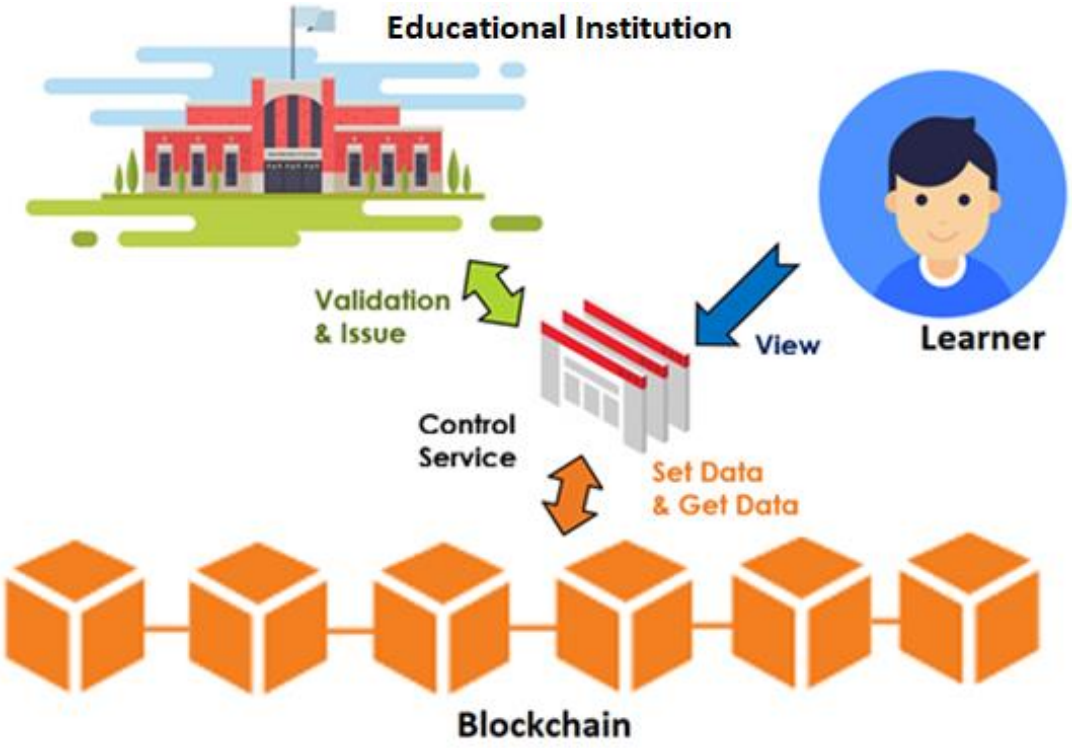


Image 15 - High-level vision

4.2 Minimal Viable Product

This dissertation has the focus in implementing a solution that can utilize smart contract to issue and store documents in the blockchain, to be later observed when requested.

A private blockchain network is configured, and to manage all the data between the users and the blockchain a Control Service is implemented, to serve as a middleware service that executes the input and output communications. The Control Service is displayed in the browser, using specific web pages to guarantee certain actions. Image 16 represents the prototype's Unified Modeling Language (UML) diagram of the Control Service functionalities.

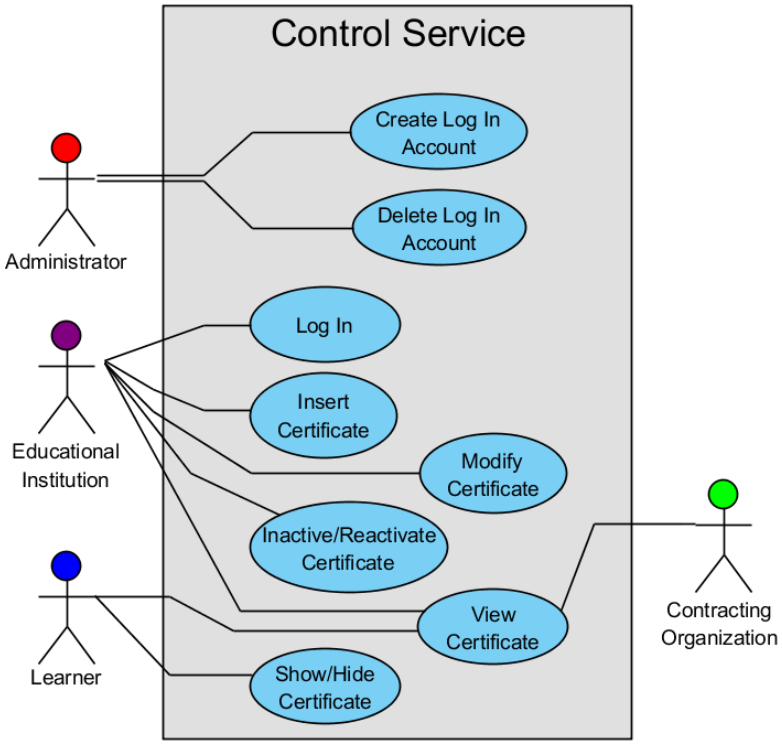


Image 16 - Prototype's UML diagram

Each individual actor is explained in the following items:

- **Learner –** An individual who has acquired knowledge and is given a certain certification. A Learner can view their certificates and manage their visibility.
- **Educational Institution –** Capable of issuing and modifying certificates in the prototype application, via smart contracts to the blockchain. These actors must log in before the manipulation of the certificates.
- **Administrator –** Maintains the operation of the blockchain and the prototype. Not a core user of the application front-end environment, however has the most important task in keeping the private blockchain network maintenance and the validation/authentication of trustful institutions to the application (trough managing account's log in).
- **Contracting Organization –** A secondary actor that can only view certificates from a provided Learner's account.

The prototype only manipulates data with the blockchain, with minimal attention regarding the front-end and back-end validations through the web pages, meaning that it is not intended to fully satisfy every possible implication of security and availability. Because the study is only revolved around the blockchain manipulation for storage and view of the smart contracts containing relevant information of the certificates.

Due to the blockchain higher maintenance costs, regarding the storage of files, the distributed network file storage service InterPlanetary File System (IPFS) is implemented in conjunction with the Ethereum blockchain, allowing to store files (mainly images), drastically reducing the costs on issuing contracts[12].

The deployment of smart contracts to the blockchain are issued by an account and the sender Ethereum account hash is authenticated using the Metamask[13] browser's plugin. In the chapter Deploy Smart Contracts is explained in more detail how the plugin works.

A database is implemented to refer the transactions hash in the blockchain, mostly to tackle the limitation regarding the search accounts that the standard blockchain operations do not provide. This usage grants a quicker search and overview of the certificates inside a certain account.

Also, the database is used to store other important information as the logins of the verified Educational Institutions accounts that can operate in the application and other information regarding the upkeep of the Control Service[14].

[12] An image file smart contract experimentation is shown in the Annex C - Base 64 Image to Blockchain. This shows the estimated costs of an issued image file to the blockchain.
[13] Additional information on Metamask plugin in the site: https://metamask.io/.
[14] More detail about the database, is explained in the chapter Database5.3.

4.3 Implementation Alternatives

The prototypes implementation for a blockchain network offers a few alternatives to ensure different types of solution for the same solution. Therefore, this project has analyzed four alternatives of implementation shown in Table 2, regarding the blockchain management.

Table 2 - Possible implementation alternatives

Alternative	Description	Advantages	Possible problems
1st	Directly inserted in the main Blockchain network	Globally distributed to the blockchain Image available everywhere	Immense Ether costs price to insert in the blockchain[15] May occur slow connections to the main blockchain nodes
2nd	Public IPFS to main Blockchain network	Globally distributed in the main blockchain network and private IPFS server Low price costs (ether), since images are held on IPFS server	May occur slow connections to the main blockchain nodes Requires both connections on blockchain and IPFS nodes IPFS public server has slow send messages Some Ether costs associated
3rd	Private IPFS to main Blockchain network	Globally distributed in the main blockchain network Low price costs (ether), since images on a private IPFS server Images are confidential on private IPFS	May occur slow connections to the main blockchain nodes Requires both connections on blockchain and IPFS nodes Can't return images if IPFS private server is offline Some Ether costs associated
4th	**Private Blockchain network and private IPFS**	**Confidentially distributed within nodes on the private blockchain network and IPFS** **Controlled administration and management to the institutions** **No real Ether costs implied**	**Available only in predefined account hashes (not global)** **Connections depend on the servers/nodes within the private network** **Requires both connections on blockchain and IPFS nodes** **Can't return the results if either IPFS or blockchain private network is offline**

The 4th alternative is most focused in the prototype development, since it has no costs for testing with the private blockchain's transactions and is the most quick and efficient way to insert an image to the blockchain using the IPFS server (immediate connection responses and controlled transaction time).

[15] Additional information in the certificate fee costs in Annex C - Base 64 Image to Blockchain.

Image 17 displays the analysis of the prototype's alternatives, based on the Analytic Hierarchy Process (AHP) method. There are four criteriums in consideration:

- **Availability -** how data is accessible
- **Cost -** represents the price cost for the transactions (in Ether)
- **Connection -** indicates how fast the responses are on the system
- **Efficiency -** specifies the overall competence of the system, in a real environment

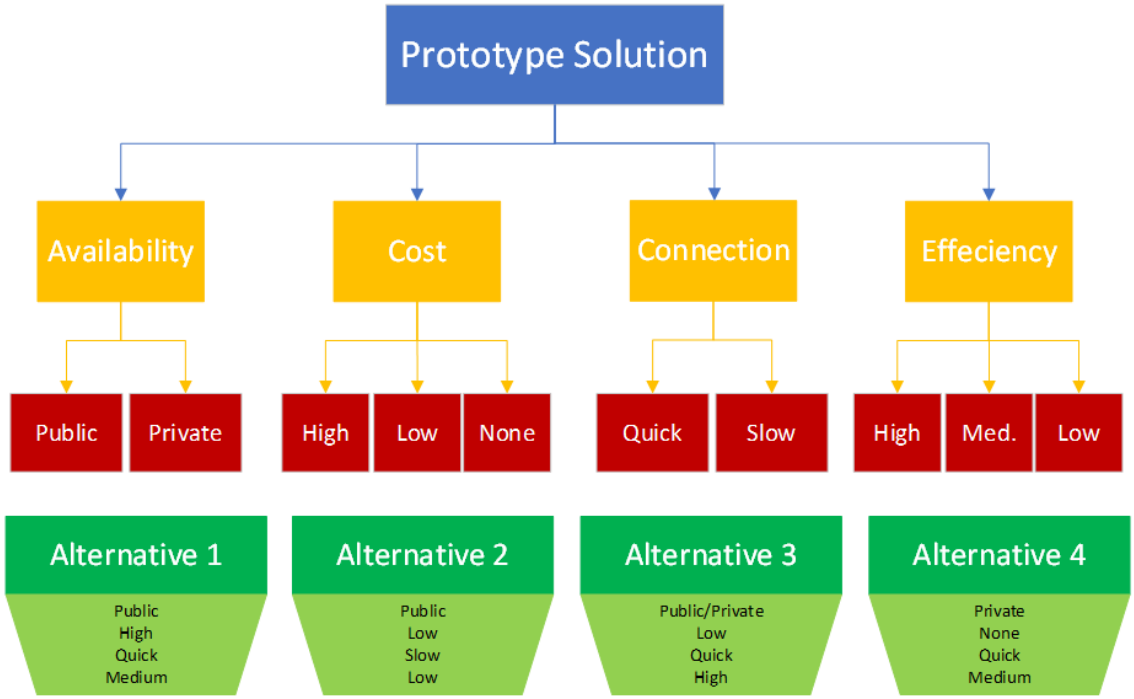


Image 17 - Prototype's AHP alternatives

4.4 Functional and Non-Functional Requirements

The functional requirements detail how the system should react to specific entries, how the user should behave in certain situations and what the system should not do. Whereas the non-functional requirements have a relevant role during the development of a system, acting as a quality criterion in the selection of a software architecture, among the various design alternatives (L. Chung & J. P. Leite, 2009).

Every core functionality of the prototype is described as a functional requirement. The prototype has to handle the certificates to the blockchain, this compromises that all the application must have the following functionalities to ensure that application:

- Account verification and authentication (described in section 5.4.1)
- Learners manage their account (detailed in section 5.4.2)
- Users view a specified account (explained in section 5.4.2.1)
- Educational Institution send and manage certificates (clarified in section 5.4.3)

Non-functional requirements are described of what the system cannot promise to do but will evaluate and test the possible outcome of those requirements. Therefore, the main non-functional requirements, evaluated in this prototype, are directly linked to the uncertain behavior of the transactions in the blockchain. Those being the **price costs** and **validation time** of the transaction. These two variables can be measured to determine a feasible outcome in the manageability (price costs) and performance (validation time) requirements.

The lowest values are intended for both variables, with a supposed limit of 2 Euros (≈ 0,0102 Ether[16]) for the price cost and 1 hour for the validation time[17]. Section 6 explains more in detail about the tests and experimentations conducted for these variables.

Other additional non-functional requirements are:

- Unauthorized access to the issued certificates in the blockchain (security requirement)
- Database and blockchain storage (capacity requirement)
- Availability of the servers/nodes (interoperability and recovery requirement)
- Administration and maintenance problems (maintainability requirement)

These additional non-functional requirements cannot be measured entirely but can possibly be present and impactful in a real production environment.

[16] Conversion value obtained from the website: https://currencio.co/eth/eur/,
[17] An assumption of what is acceptable for the general individual, in this document's author perspective.

4.5 Domain Model

The Domain Model is represented in the Image 18, followed by the explanation of each entity in Table 3.

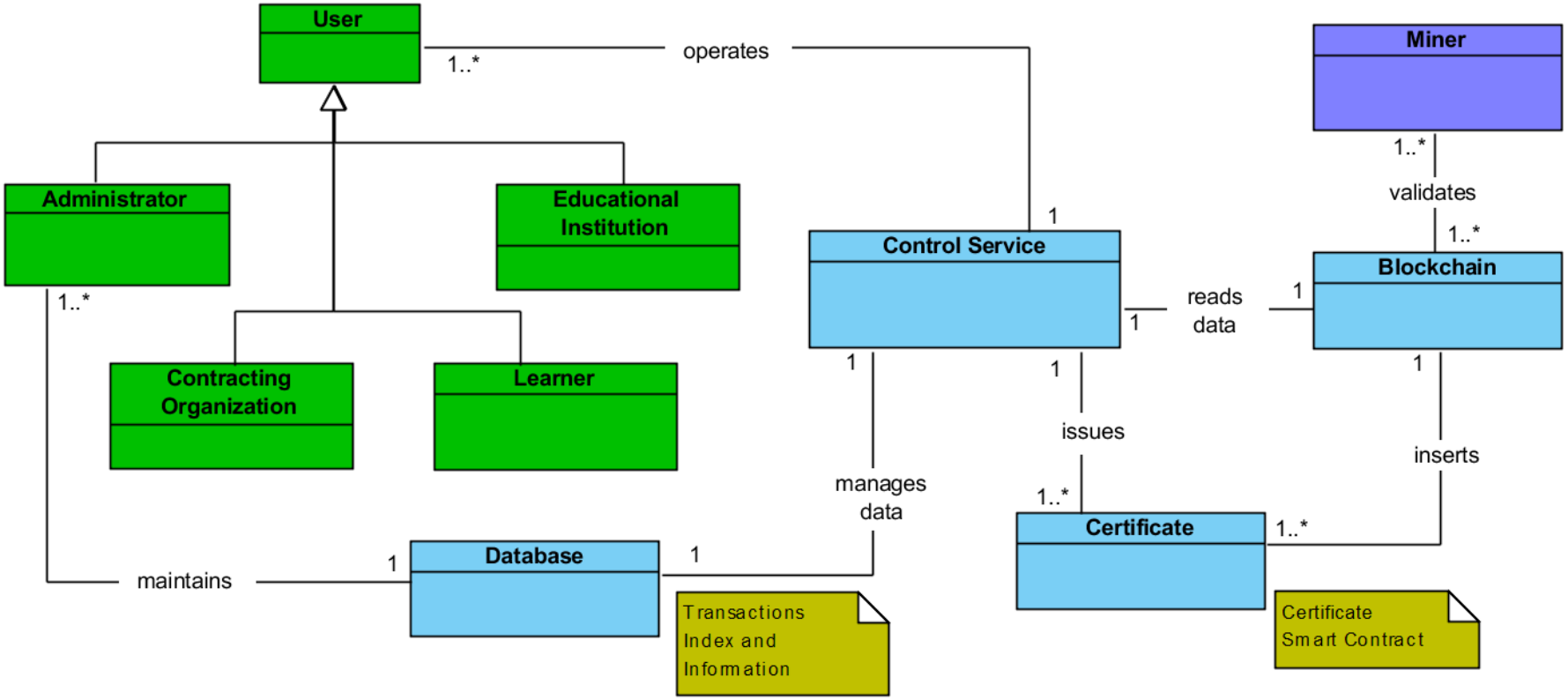


Image 18 - Prototype's Domain Model

Table 3 - Domain Model's entities description and relation

Entity	Description	Relation
User	Group of entities that are represented in the application	Operates Control Service
Administration	User that has the administrator permissions to maintain the database information	Maintains Database
Educational Institution	User that can manage the certificates issued by the educational institution	
Learner	User that has the certificates associated to his account	
Contracting Organization	User that can only view the Learner's certificates	
Control Service	Main entity that operates with all the development's processes	Reads data in Blockchain Issues Certificates Manages data in Database
Certificate	Entity representing the Certificate smart contract	Inserts in Blockchain
Database	Data storage entity	
Blockchain	Entity that characterizes the blockchain server (with Transactions and Blocks)	
Miner	Process entity that operates in the blockchain for validating blocks	Validates Blockchain

4.6 Modeling the Business Process

Modeling the Business Process is better described using Business Process Model and Notation (BPMN) since it represents an illustration of a group of activities carried out in a logical sequence, with the objective of producing a demonstration of the service's flow.

BPMN is exemplified in the Image 19. The image represents a series of pools (rows), that are the main entities of the blockchain interaction:

- Learner
- Educational Institution
- Blockchain, composed of:
    - Educational Institution Account
    - Transactions
    - Learner Account
    - Miner

The purpose of the flow is to demonstrate all the interaction between all the pools, with the result being the completely issue of the certificate (and verification) to the blockchain.

The interaction begins by the Learner that requests the certificate to the Educational Institution, that acknowledges the Learner and asks the account hash, in which the Learner wants to allocate the certificate. After receiving the account, the Educational Institution fills a form and issues the certificate to the blockchain, as a Transaction. This Transaction is now pending for validation/insertion to the blockchain by some Miner. When the Transaction is validated, inside de blockchain, the Learner can check the certificate inserted to the Learner's account.

Image 19 has a secondary representation of the Miners, which demonstrate the general concept of their job on the blockchain. Even if is not directly linked with the certification issued by the Educational Institution, the continuous validation of the transactions must always be engaged by the Miners, blockchain's automated processes.

Miners await the transaction issued by the Educational Institution and then checks the transaction, allocates the transaction to a block to insert to the blockchain, once the block hash is complete.

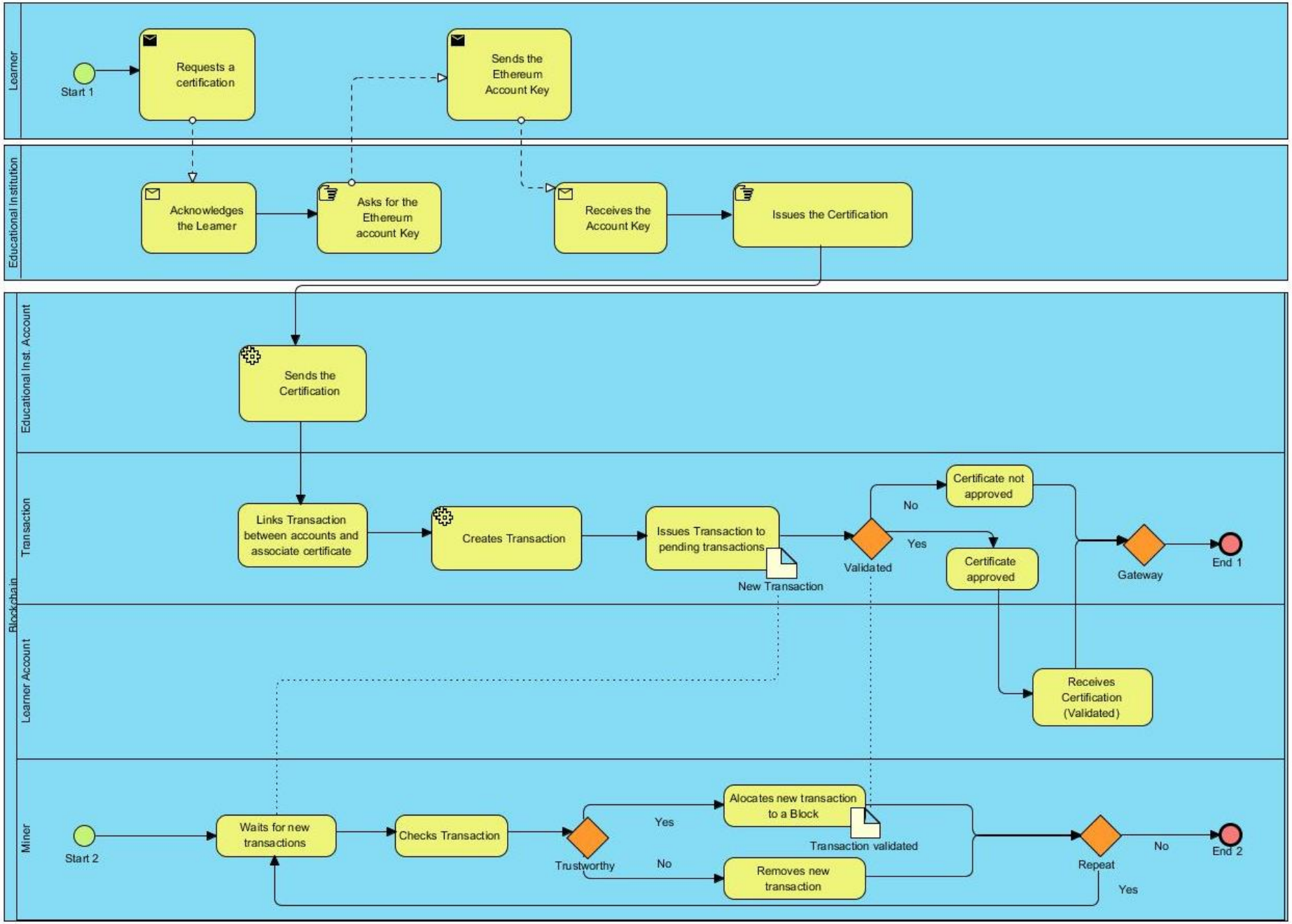


Image 19 - Project's BPMN

4.7 Development Comparison

Similarities to this prototype implementation are the Blockcerts (by MIT Media Lab) and the TrueRec (by SAP), explained in the previous section Blockchain application in education.

Both projects offer the issue of documents to the blockchain, and afterward the documents can accept by the recipient's account, adding them to their account. These applications promote verification, validation, accessibility and privacy of the inserted information.

A mobile application is equally implemented in each project, that allow to easily view and share the relatable information of the documents, in the portable device. The sharing of the documents is displayed in the browser, by an obtainable Uniform Resource Locator (URL) of the person's account.

There isn't any relatable difference between these applications since the purpose is the same: certifications are store inside the public blockchain and are displayed in the user's account, with an advantage in delivering a trustful and verified information.

Comparing these implementations to the developed prototype, the features that aren't implemented: the mobile application, the e-mail document acceptance and the public blockchain accessibility.

The non-implementations of those features in the prototype is intentional, since the only purpose is to issue smart contracts to private blockchain, and those features are more of an extra, improving the accessibility, portability and acceptance of the documents.

5 Implementation

The Implementation chapter is subdivided in a series of procedures to accomplish the project's design. In each section has the explanation of the processes that were necessary to be followed, to obtain the desirable solution.

5.1 Installation

For the environment set-up (testbed), this project is implemented in a separate virtual machine - Linux Mint[18]. This operating system is installed using the Oracle VM VirtualBox and has the systems requirements of 32GB of storage with a 4GB RAM.

As for the software installed on the machine, the following items show all the required software that were installed:

- **Geth -** Allows to create and manage the Ethereum blockchain network inside a machine and can also be used to interact with the main Ethereum blockchain network (id = 1). However, in an Ethereum main network requires having an account with ether to send transactions, to be later validated and inserted permanently to the main blockchain network.
- **IPFS -** A developed protocol and network to allow the fast exchange of files within a distributed network. Supports file storage and for certificate image storage is a viable choice to implement. A distributed network file storage is essential for the certificate's image storage, another alternative is using the main blockchain, but as referred previously it comes with a large cost price in maintaining the images in the blockchain.
- **MySQL -** An open source database that ensures the database installation inside a machine, to purposely storage all the information regarding the prototype, so the control service can keep track of all smart contract's transactions inserted to the

[18] Linux Mint is a free open source operating system, that offers an easy to use interface with all the Linux utilities. More info in the Linux Mint home page: https://linuxmint.com/.

blockchain, and other useful information. In the Database section shall be explained the structured database.

- **Node.js[19] –** A platform for developing high-performance scalable web applications using *Javascript* code.
- **Express.js[20] –** Node.js framework that sets the route abstractions, *middlewares* and other functions to facilitate the creation of Application programming interface (API).

The installation and configuration of the software: Geth, IPFS and MySQL used in the prototype is detailed in the Annex D - Software Installation and Set-up.

5.2 Developed Smart Contracts

This project is represented with three different types of developed Smart Contracts to manage all the necessary information to the control service, regarding the certification and login management (insertion, modify and delete[21]).

Each smart contract constructor implies a new creation of the smart contract to the blockchain, as a transaction, and the other functions creates a transaction reference to that smart contract. These transactions have price costs and validating time implied to them.

These smart contracts are developed with the authorization pattern on the Ownership Pattern (M. Wöhrer & U. Zdun, 2018), for the validation of the account that issued the smart contract, as an owner. Allows only the owner to alter the content of the smart contract, so that other accounts can never tamper with the information.

The developed smart contracts are Certificate, with two development concepts, and the optional Login. All will be explained separately in the following subchapters.

5.2.1 Certificate

The smart contract that is implemented in the prototype solution to manage the insertion, modification and inactivate/reactivate processes. This contract fully developed to ensure the utilization of the processes that are required for managing the certificates.

To link a learner's account, the educational institution must always set the send account hash in the smart contract.

This smart contract type operates with the educational institution account and allows to insert a reference of an image hashed, using the IFPS server. The hash allows the reference to the IFPS

[19] More information about Node.js on the web page: https://nodejs.org/en/
[20] Additional information about Express.js on the web page: https://expressjs.com/
[21] Delete contract is a utilized conventional term to refer the contract active status. Since all the information inside the blockchain is immutable, the active status is referred as a deleted contract.

network, retrieving an image file that can be viewed in the browser, using the control service web page.

In addition to the IFPS image hash, the Certificate also stores information regarding the certificate in a text format (description).

It is possible to later define the state of the contract, with the active variable. This means that if the smart contract is not active it shall be ignored by the control service, to display the information on the browser.

The variables inside the Certificate smart contract are allocated as public[22] and are responsible to store the following information:

- **contractOwner –** address of the sender account hash
- **sendToAccount –** refers to the contract of the specified the learner's account hash
- **ipfsHash –** represents the hash of the image in the IFPS
- **description –** optional parameter referring to a description that can be shown in the contract
- **active –** represents the status of the contract, as a "true" or "false". When first inserted the status is represented as "true", if later deleted the new contract is associated with a "false" active status.

For the certificate management, the following functions are implemented in the smart contract:

- **constructor (_sendToAccount, _ipfsHash, _description) –** creates a new certificate based on the parameters. The contractOwner is defined with the sender account hash and the active is predefined as "true".
- **setContract (_sendToAccount, _ ipfsHash, _description) –** modifies the contract variables with the specified parameters.
- **setActive (_active) –** specifies if the contract is active, if "false" the contract is deleted (can recover/reactivated contract with the "true" value).

The Code 1 represents the smart contract of the certificate, with all the specified information detailed in this section.

[22] Variable with a public statement refers that the value can be obtained, without the need to create a get function for each variable.

```
pragma solidity ^0.4.24;
contract Certificate {
    address public contractOwner;
    address public sendToAccount;
    string public ipfsHash;
    string public description;
    string public active;

    constructor(address _sendToAccount, string _ipfsHash, string
_description) public {
        contractOwner = msg.sender;
        sendToAccount = _sendToAccount;
        ipfsHash = _ipfsHash;
        description = _description;
        active = "true";
    }

    function setCertificate(string _ipfsHash, string _description) public
{
        if (msg.sender != contractOwner) { revert(); }
        ipfsHash = _ipfsHash;
        description = _description;
    }

    function setActive(string _active) public {
        if (msg.sender != contractOwner) { revert(); }
        active = _active;
    }
}
```

Code 1 - Certificate (Smart Contract)

5.2.2 Certificates (alternative)

This smart contract is an alternative representation of the Certificate, that follows the same concept, however has a different structure, in which is implemented as a certificate storage to all the issue certificates.

It has a more complex structure, to the meaning that the smart contract only needs to be issued once to the blockchain and modify its information, in creating new certificates or modifying the existent certificates.

As a development stand point, the certificate alternative is only issued once per educational institution, and for every added new certificate the index, that can be later be referred to obtain a specific certificate. The length of this smart contract increases as long new certificates are inserted.

Comparing this smart contract to the generic Certificate smart contract (previous Certificate smart contract), it has an advantage that is to issue only once the contract information, meaning that it costs less comparing to the previous smart contract.

However, has the disadvantage in the continuous growth of the smart contract (non-stop insertion of new certificates to the smart contract), that is greatly time-consuming task if someone tries to scan the smart contract, without knowing the right reference.

This disadvantage is the meaning why the generic Certificate is used instead of the alternative Certificate.

The variables inside the alternative Certificate smart contract are responsible to store the following information:

- **contractOwner –** address of the sender account hash
- **lastID –** states the last ID index of the smart contract
- **Certificate –** a structure that allow to store the following variables in a single certificate (same variables as the generic Certificate):
    - **sendToAccount –** refers to the contract of the specified the learner's account hash
    - **ipfsHash –** represents the hash of the image in the IFPS
    - **description –** optional parameter referring to a description that can be shown in the contract
    - **active –** represents the status of the contract, as a "true" or "false". When first inserted the status is represented as "true", if later deleted the new contract is associated with a "false" active status.
- **certificates –** saves all the Certificate's variables
- **certificateAccounts –** indexes all the variable *certificates* with their respected ID

As for the alternative Certificate management, the following functions are implemented in the smart contract:

- **constructor() –** creates a new alternative Certificate smart contract, with the contractOwner defined with the sender account hash, and without any new certificate.
- **newCertificate(_sendToAccount, _ipfsHash, _description) –** creates a new certificate base on the specified parameters. This new certificate receives the index ID of the lastID variable
- **setCertificate(_ID, _sendToAccount, _ipfsHash, _description) –** to the Certificate with the _ID, modifies the contract variables with the other specified parameters.
- **setActive(_ID, _active) -** to the Certificate with the _ID, specifies if the contract is active, if "false" the contract is deleted (can recover/reactivated contract with the "true" value).
- **getCertificate(_ID) -** obtains the Certificate information of the indexed _ID. Returns all the information inside the searched Certificate

Code 2 represents the smart contract of the alternative certificate, with all the specified information detailed in this section.

```
pragma solidity ^0.4.24;
contract InstitutionCertificates {
    address public contractOwner;
    uint public lastID;

    struct Certificate
    {
        address sendToAccount;
        string ipfsHash;
        string description;
        string active;
    }

    mapping (uint => Certificate) certificates;
    uint[] public certificateAccounts;

    constructor() public {
        contractOwner = msg.sender;
    }

    function newCertificate(address _sendToAccount, string _ipfsHash, string
_description) public {
        if (msg.sender != contractOwner) { revert(); }
        lastID = lastID + 1;
        Certificate l_certificate;
        l_certificate = certificates[lastID];
        l_certificate.sendToAccount = _sendToAccount;
        l_certificate.ipfsHash = _ipfsHash;
        l_certificate.description = _description;
        l_certificate.active = "true";
        certificateAccounts.push(lastID) -1;
    }

    function setCertificate(uint _ID, address _sendToAccount, string _ipfsHash, string
_description) public {
        if (msg.sender != contractOwner) { revert(); }
        Certificate l_certificate;
        l_certificate = certificates[_ID];
        l_certificate.sendToAccount = _sendToAccount;
        l_certificate.ipfsHash = _ipfsHash;
        l_certificate.description = _description;
    }

    function setActive(uint _ID, string _active) public {
        if (msg.sender != contractOwner) { revert(); }
        Certificate l_certificate;
        l_certificate = certificates[_ID];
        l_certificate.active = _active;
    }

    function getCertificate(uint _ID) view public returns (address, string, string,
string) {
        return (certificates[_ID].sendToAccount,
                certificates[_ID].ipfsHash,
                certificates[_ID].description,
                certificates[_ID].active);
    }
}
```

Code 2 - Alternative Certificate (Smart Contract)

5.2.3 Login (optional)

Login smart contract contains the accounts authorized to login, validating the access of the Educational Institution that can operate in the control service for certificate management.

This contract was initially implemented, however it was removed from the prototype, since it not essential the login through stored smart contract in the blockchain, as it comes with some unnecessary fee cost to maintain them in the blockchain. The best implemented approach comes from the usage of the Metamask plugin and the database's Login table23 to authenticate the blockchain accounts, on the client-side. There is always the need to always use Metamask to confirm the transactions issued by an account, so the Login smart contract has a needless usage, since the Metamask plugin could authenticate the Educational Institution's account.

Nevertheless, this login implementation can be an optional way to apply in a private blockchain network, as the ether costs aren't a concern, because the storage costs are maintained by the private network nodes.

Login smart contracts is responsible to store the information:

- **accName –** Name of the educational institution
- **loginName –** Defined login name (SHA3)
- **loginPassword –** Defined login Password (SHA3)
- **active –** Specifies if the account is active

The loginName and loginPassword has its information converted to a Secure Hash Algorithm (SHA) for a secure approach to the authentication process, to with only the user who knows the variable unhashed content can be authenticated and modify the contract content.

Consequently, produces a different approach from the authorization pattern on the Ownership Pattern, to which the authorization is done by the account's credentials (loginName and loginPassword), instead of the owner of the smart contract. Meaning that anyone who knows the exact loginName and loginPassword can alter the smart contract content.

This smart contract has the following functions:

- **constructor(_accountHash, _accountName, _loginName, _loginPassword) -** creates a new smart contract with the specified parameters, and actives the contract ("true")
- **openLogin(_loginName, _loginPassword) –** returns an boolean value if the login was authenticated. The arguments are passed to the smart contract as words, converted to SHA3 and compared to the variables available on the smart contract. If returned "true" the login the variables are equal SHA3 and has been successful authenticated; otherwise, if returned "false" the variables weren't equal, so the authentication wasn't successful

[23] In the Database section explains the Login table

- **changeLogin(_loginName, _loginPassword, _accountName, _newLoginName, _newLoginPassword) -** allows to change accountName, loginName and loginPassword, only if the supplied _loginName and _loginPassword equals the corresponding variables on the smart contract (same authentication process as the openLogin function)
- **setActive(_loginName, _loginPassword, _active) –** changes the active state to "true" or "false". Has the same authentication process as the previous functions.

The Code 3 shows the developed code of the Login smart contract.

```
pragma solidity ^0.4.24;
contract Login {
    address public accountHash;
    string public accountName;
    bytes32 loginName;
    bytes32 loginPassword;
    string public active;

    constructor(address _accountHash, string _accountName, string
_loginName, string _loginPassword) public {
        accountHash = _accountHash;
        accountName = _accountName;
        loginName = sha3(_loginName);
        loginPassword = sha3(_loginPassword);
        active = "true";
    }

    function openLogin(string _loginName, string _loginPassword) constant
public returns (bool) {
        if ((loginName == sha3(_loginName)) && (loginPassword ==
sha3(_loginPassword))) {
            return true;
        }
        return false;
    }

    function changeLogin(string _loginName, string _loginPassword,
string _accountName,  string _newLoginName, string _newLoginPassword)
public {
        if ((loginName == sha3(_loginName)) && (loginPassword ==
sha3(_loginPassword))) {
            accountName = _accountName;
            loginName = sha3(_newLoginName);
            loginPassword = sha3(_newLoginPassword);
        } else { revert(); }
    }

    function setActive(string _loginName, string _loginPassword, string
_active) public {
        if ((loginName == sha3(_loginName)) && (loginPassword ==
sha3(_loginPassword))) {
            active = _active;
        } else { revert(); }
    }
}
```

Code 3 - Login (Smart Contract)

5.2.4 Deploy Smart Contracts

For every implemented smart contract, it is necessary to deploy them to the blockchain, so it can take it desirable effect in storing and managing the smart contract information for the prototype. The deployment of smart contracts is done via the browser, using the implemented automatic procedures for the management of certificates in the Control Service.

However, an Ethereum account needs to be unlocked[24] so the account can deploy the smart contracts' transactions to the blockchain, authenticating the account in the blockchain networks. Since the browser doesn't offer a direct approach of implementing the account unlock on the client-side, the Metamask plugin is used for the account unlock.

Metamask plugin serves as a connection of identifying user's Ethereum accounts inside various configured blockchain network, through the identification of their account's privatekey or either their validated account hash and the password. After the correct association of the user account, the smart contract is now ready to be deployed onto the blockchain network.

Image 20 shows how the Metamask plugin associates the user's account, and Image 21 is a Metamask confirmation message when a transaction has been issued from the account.

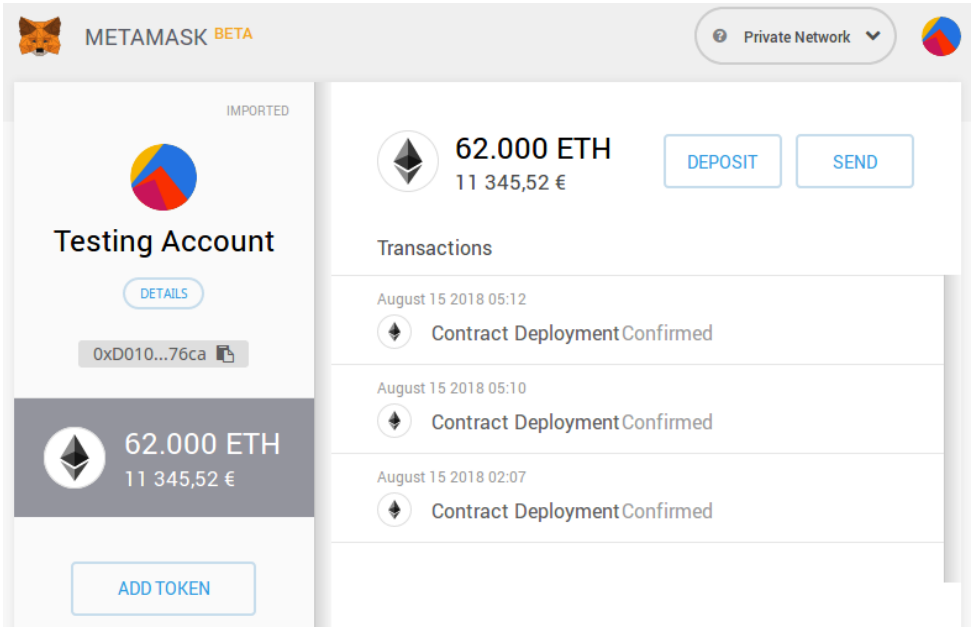


Image 20 - Metamask's account association

[24] Unlocking an account refer to the confirmation on Ethereum account's credentials. If the account is unlocked, the program can ensure transactions to the blockchain, using that account as a validated sender.

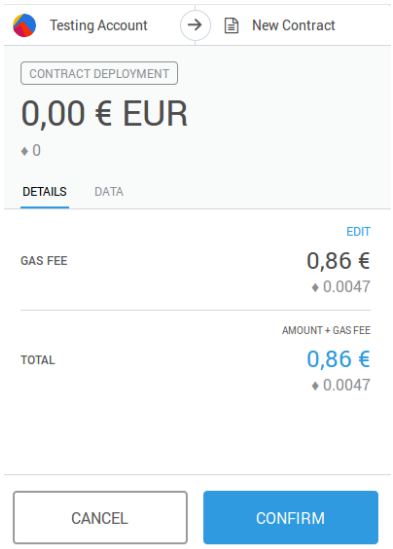


Image 21 - Metamask's transaction confirmation

The deployment of the smart contract is executed on the client-side, as the Metamask authenticates the account, therefore the contract needs to be issued in the full state the first time it is inserted, after the issued contract functions can be called to obtain the values or change the variable values (mostly has a *set* prefix).

Each time a *set* prefix function is called, it is necessary to issue and confirm the transaction to the blockchain. These operations require ether cost fees to execute, but typically in a much smaller fee cost comparing to the full smart contract issue, this is due to smart contract code has already been deployed to the blockchain.

To better clarify the previous statement, the Image 22 shows the cost comparison between two transactions: the first transaction is the full smart contract and the second transaction is a set prefix call, on the first transaction.

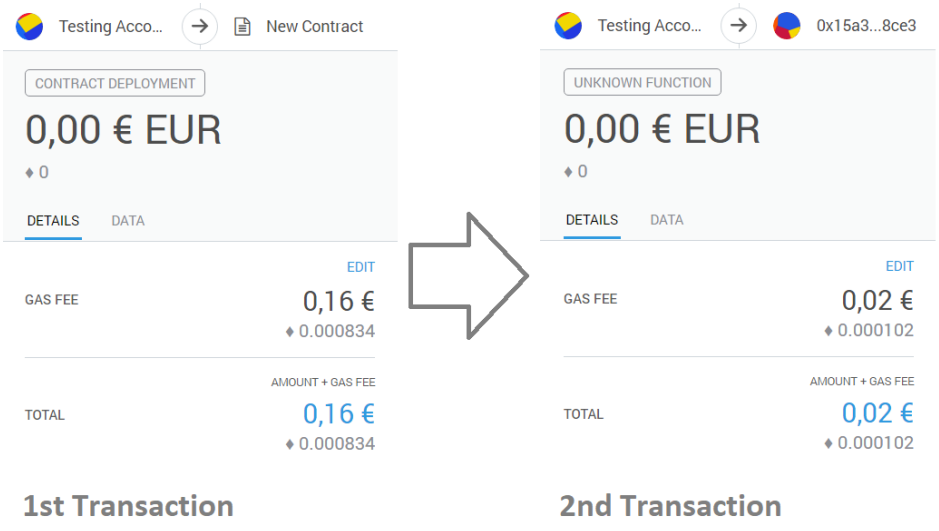


Image 22 - Transaction's cost comparison

In the Metamask contract confirmation, there is an *EDIT* option where it is possible to edit the contract's cost fee. This is mainly to control the validation time of the contract, to determine how fast the contract is validated associated to the contract high-cost fee[25].

Image 23 shows the customization of gas in the Metamask plugin, where the *Gas Price* is the amount of ether the user is going to pay for each unit of gas and *Gas Limit* refers to the maximum amount of gas that the user is willing to spend on the transaction (C. Zorzini, 2018). Both values can be adjusted, however Metamask plugin adjusts these values, based on the network estimated success rate in the insertion of a transaction to the blockchain.

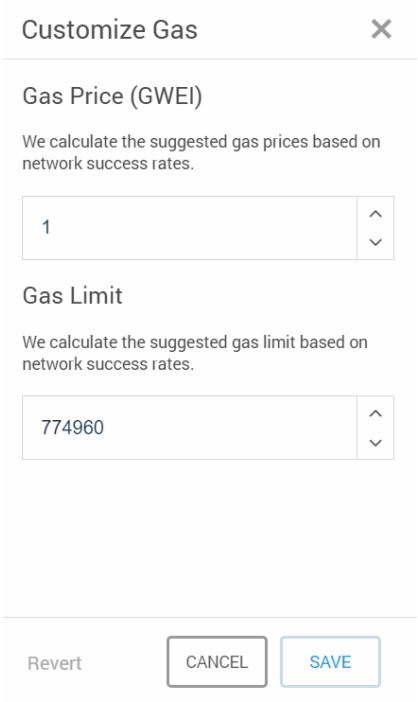


Image 23 - Metamask's Customize Gas

[25] Transaction's mining always takes priority in transaction with a higher *Gas Price*.

5.3 Database

The development of a database in the prototype was intended with the main purpose in storing all the inserted transactions hash, to serve as an index, keeping track of the transactions inside the blockchain network that has the necessary information for the prototype's execution. This is due to the blockchain's limitation in returning all the transactions inside a certain account.

Retrieving all the account's transactions is necessary for the project's development, since the accounts need to list all the available Certificate smart contracts of a specific account.

The database also preserves the information of all the verified institutions to login in the prototype application, for the certificates management.

5.3.1 Data Model

The Data Model in Image 24 shows the database tables, fields and relations and the following Table 4 has the description on each respective table and fields.

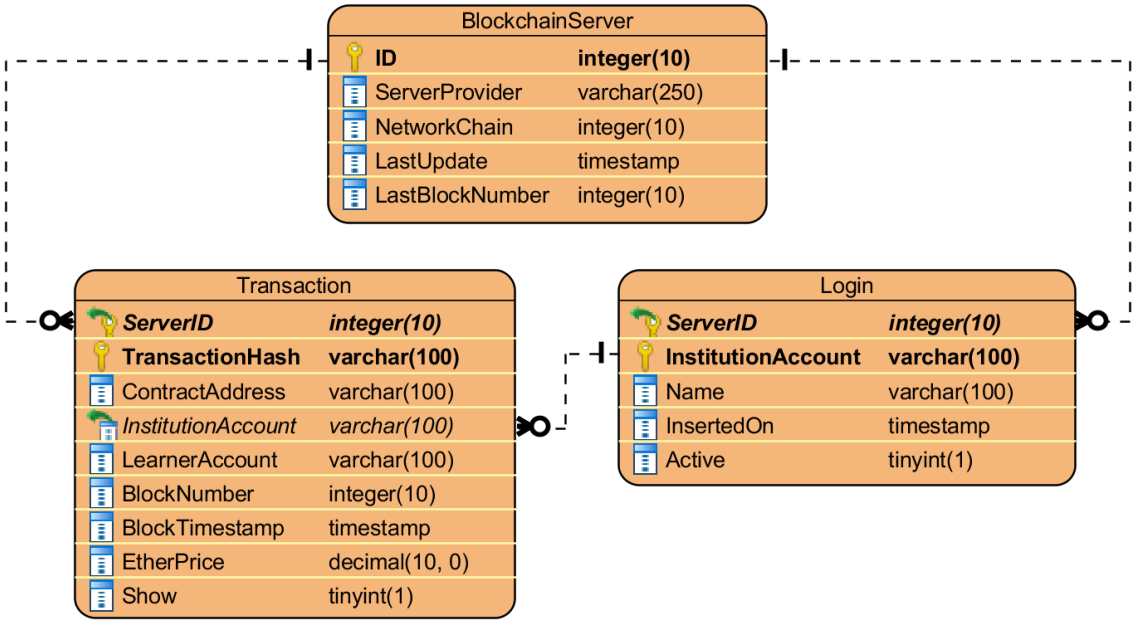


Image 24 - Prototype Data Model

Table 4 - Data Model's fields explanation

Table	Field	Description
BlockchainServer		**Stores the blockchain server information used in the prototype, as well as the last updated blocks in the routine**
	1. ID	• Incremented primary key number, representing the Blockchain Server identification on the database
	2. ServerProvider	• The link to the blockchain network server, with the respective port number
	3. NetworkChain	• Blockchain network chain number
	4. LastUpdate	• Timestamp of the last update time and date that the update routine was executed
	5. LastBlockNumber	• Last block number that the update routine was executed
Transaction		**Provides a storage to all the transactions necessary information that are used in the prototype application**
	1. ServerID	• Primary and foreign key that refers the BlockchainServer where the transaction is assigned
	2. TransactionHash	• Primary key to refer the transaction hash in the blockchain network
	3. ContractAddress	• Smart contract reference, inside the transaction
	4. InsitutionAccount	• Account of the Educational Institution that has sent the transaction
	5. LearnerAccount	• Account of the Learner that has received the transaction
	6. BlockNumber	• Block number of the blockchain network that the transaction is inserted
	7. BlockTimestamp	• Timestamp when the transaction/block was validated and inserted to the blockchain network
	8. EtherPrice	• Cost that was applied to the transaction when it was inserted to the blockchain
	9. Show	• Boolean that manipulates the view of the transaction in the application. If true it is visible on the browser, otherwise it is not shown
Login		**Serves as a storage to all the permitted institutions to log in and manage the certificates inside the prototype application**
	1. ServerID	• Primary and foreign key that refers the BlockchainServer where the transaction is assigned
	2. InsitutionAccount	• Account of the Educational Institution that is allowed to log in and manage the certificates
	3. Name	• Name of the Educational Institution
	4. InsertedOn	• Timestamp of when the login record was inserted
	5. Active	• Boolean that states if the login record is active or not. If active can login and manage the certificates inside the prototype, otherwise it isn't allowed to login and manage the certificate.

5.3.2 Update Routine

The Ethereum blockchain network doesn't have a default algorithm to search all the transactions in a specific account, creating a limitation on the blockchain network in obtaining all the transactions of an account. Therefore, the Update Routine has the purpose of inserting and updating data to the prototype's database, based on the available information inside the blockchain, maintaining the tables in the database revised and organized each time the update routine executes.

Based on *Javascript* code, the Update Routine reviews every block, and their respective transactions, to obtain a slice of hash that allows compare to the Certificate smart contract initial state and validate if the transaction's contract is a Certificate smart contract.

The slice of hash comparison is represented in the Image 25, where the Contract Hash represents the prototype's smart contract and the other blockchain transactions are compared with that Contract Hash. If the Contract Hash is equal to the first part of the transaction's contract hash, the transaction reference is added to the database, otherwise it is ignored.

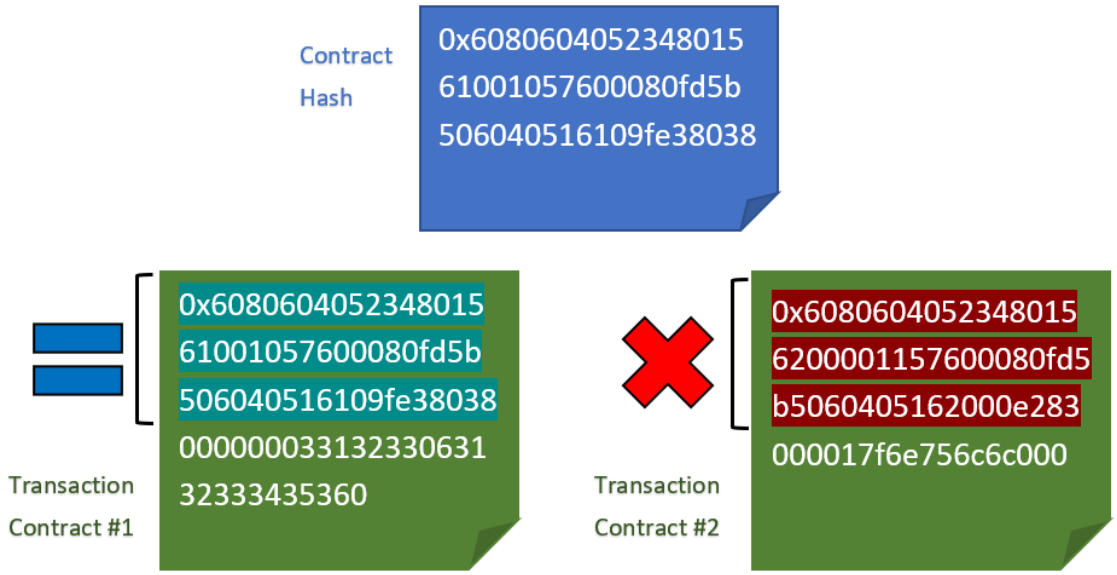


Image 25 - Contract hash evaluation

The process uses a sequential flow, that is represented on the Flow Chart in the Image 26.

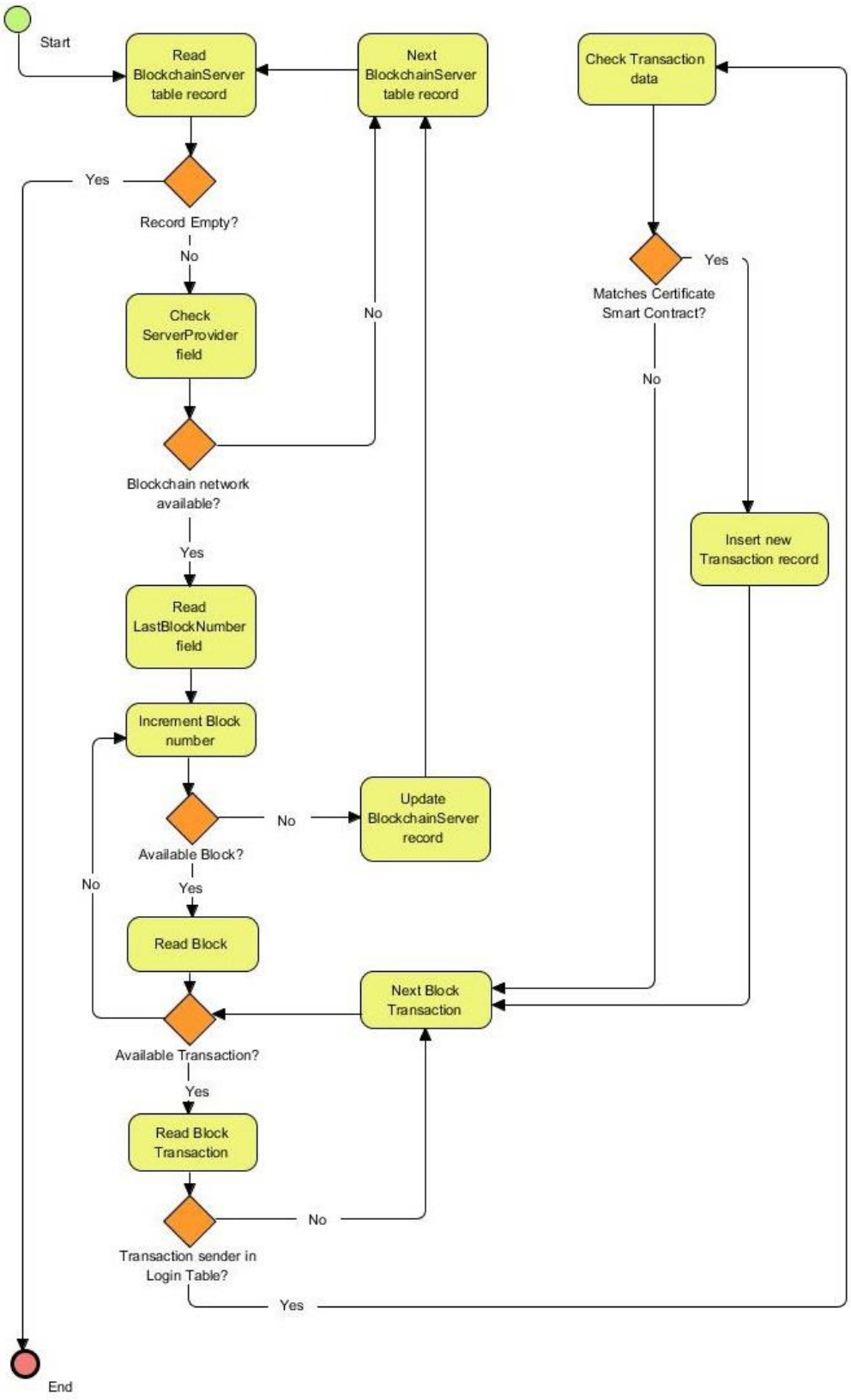


Image 26 - Update Routine flowchart

The update routine is composed by the next steps that follow the explanation of the Flowchart:

1. Firstly, scans the records on the table BlockchainServer to determine the ServerProvider. Accesses the blockchain's ServerProvider and verifies if the blockchain is accessible; if not scans the next BlockchainServer record
2. If the blockchain network is available, obtains the information of the last updated block (LastBlockNumber) on the table BlockchainServer, in the selected record
3. Next checks each block with their transactions, starting on the last block number, and verifies if the sender account on the transaction is an Educational Institution, available on the Login table
4. If the previous condition is true, verifies the slice of the contract address in the transaction, if it corresponds to the Certificate smart contract this transaction is added to the Transaction table, with all the necessary information. Otherwise checks the next block on the blockchain
5. When all blocks are verified, the routine updates the BlockchainServer table with the last block number (LastBlockNumber) and the current timestamp, date and time, of the system (LastUpdate)

Execution of the update routine has the output of the transactions added to the database. This output is shown in the console, in Image 27, where the information regarding the transaction reference is inserted as a record in the Transaction table.

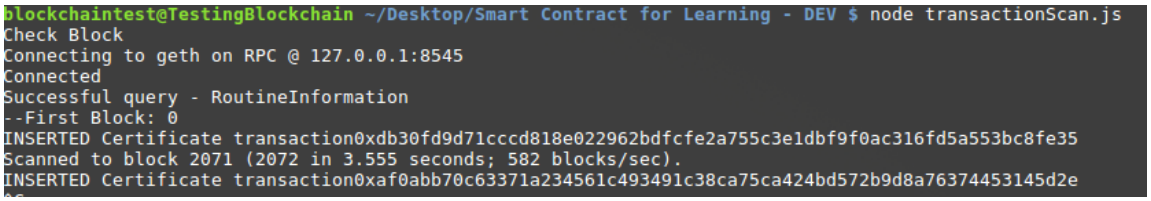


Image 27 - Update Routine console output

5.4 Control Service

This section documents the explanation of the Control Service development, that interacts with the specified blockchain network, database, IPFS and serves as a user interface on the browser, implemented using HTML and *Javascript* (with Node.js[26] and Express.js[27]) code.

The Control Service application allows: Educational Institution to manage the certificates in the insert, modify, delete and reactivate statements; Learner to view and manipulate the visibility of their account's certificates; and the other users (Contracting Organization) only view the visible certificates in the browser. As for Administrator, their purpose is to maintain the Control Service operational and allocate new verified logins to the database Login table.

For most interactions within the Control Service, the Metamask plugin is required to be installed in the browser, with a verified account, to grant Educational Institution permissions on the authenticated account.

The following sub-sections explain the prototype application execution for each important procedure, and the prototype demonstration can be found in the Annex F - Prototype Demonstration, to visually complement how the development is structured to each sub-section.

[26] Node.js is a platform for developing high-performance scalable web applications using JavaScript. More information on the web page: https://nodejs.org/en/

[27] Express.js is a Node framework that creates route abstractions, *middlewares* and other functions to facilitate the creation of Application programming interface (API). More detail about express on the web page: https://expressjs.com/

5.4.1 Account Hash Verification

For authenticating the users on the web application, with their respective account, the account is verified using the browser plugin Metamask, allowing the server to know which account is specified on the blockchain. Operations with the Learner's certificates view manipulation and Educational Organization's certification management requires the verified account through Metamask.

The purpose of the browser plugin is to authenticate the blockchain account hash and pass the smart contracts issued by the account to the blockchain, grating the web application the ability to deploy smart contracts. Since it has an indispensable usage, the account login authentication can utilize this plugin function, in verifying the blockchain account, to grant permissions to the web application. This login authentication is automatic once a Metamask account is selected.

However, the accounts inside the blockchain do not directly specify each user's role, this is main reason the Login table was created in prototype database with the field InstitutionHash, to assign the verified Educational Institutions. This allows the prototype to know if blockchain user account has the permission to manage certificates.

As for the certificates view, only the authenticated Learner account, on the Metamask plugin, can show or hide the certificates (manage view). This doesn't need a special grant, equal to the storage of the Educational Institutions accounts inside de database, nevertheless all the Learners accounts are assigned in all the Transaction's table records for filtering measures[28].

For other users that only want to view the certificates on a specified account (for example Contracting Organizations), the Metamask plugin isn't required, only the account hash is necessary to list all the certificates in that specified account.

[28] Filter records to obtain a list of the issued transactions in a specified Learner account.

5.4.2 Certificate Visibility

The certificates, shown on the browser, are determined with the variables retrieved from the blockchain Certificate smart contract and are displayed on the web page as:

- **Image File –** shown image file, retrieved from the IPFS server
- **Certificate Hash (IPFS) –** image hash from the IPFS server
- **Description –** text description of the certificate
- **Contract Hash –** smart contract hash address on the blockchain

The Image 28 represents a certificate example shown on the browser, where the information is extracted from the blockchain and the image file from the IPFS. Additional certificates are represented in the list, order by date.

Image 28 - Issued Certificate (example)

5.4.2.1 View Certificates

The View Certificates displays a list of the available certificates in one account[29]. Processes by scanning every issued Certificate smart contract from the blockchain, with the help of the Transaction table index reference, to obtain and show all the certificates.

This process can be executed by any user that has access to the web application and has an automatic display of all the available certificates if the user has the Metamask account authenticated. However only retrieves the certificates found in the Transaction table, that refers to the transactions found on the blockchain. If no certificate is found, the user can choose to fill the account hash, available on the web page, to list all the shown[30] certificates accessible in the account.

The process flow is represented in the Image 29.

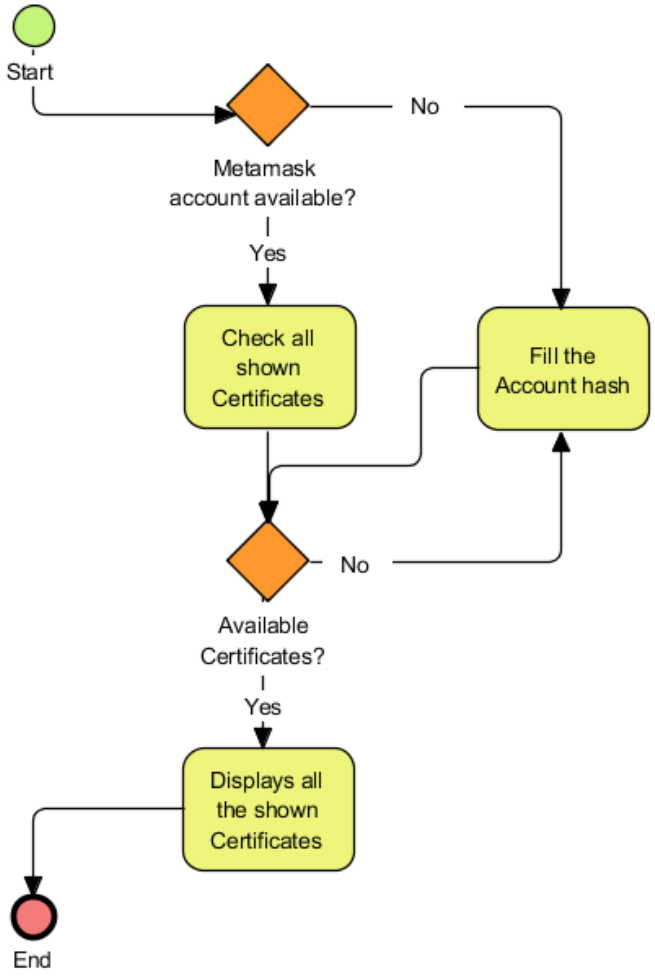


Image 29 - View Certificate flowchart

[29] If the account is an Educational Institution, then displays all the smart contracts sent by the Educational Institutions

[30] Only lists the certificates that have the field Show active in the Transaction table

The flowchart representing the certificate view has the following steps:

1. Metamask account validation, if not authenticated the account hash is required to be filled. Otherwise, the application automatically assumes the account on the Metamask
2. The application checks all the available Certificate smart contracts in the Transaction table, if no Certificate found, shows a message requesting another account hash
3. If certificates are found in the table, the web application fetches the information on the blockchain for each transaction's Certificate smart contract variables
4. After the certificate information is retrieved from the blockchain, the image hash is used to get the image file from the IPFS server
5. When there were found certificates available, the control service displays a certificate list, based on the information retrieved from every certificate

5.4.2.2 Manipulate Visibility

The Manipulate Visibility is a process that can only be manipulated by the Learner that has the blockchain account authenticated on the Metamask. This authentication allows to check if the specific account belongs to the Learner.

When a list of certificates is retrieved, the user can then manipulate the certificates visibility to show or hide specific certificates in their account. All these changes are later updated to the Transaction table.

The flowchart in the Image 30 represents the flow of the Manipulate Visibility process.

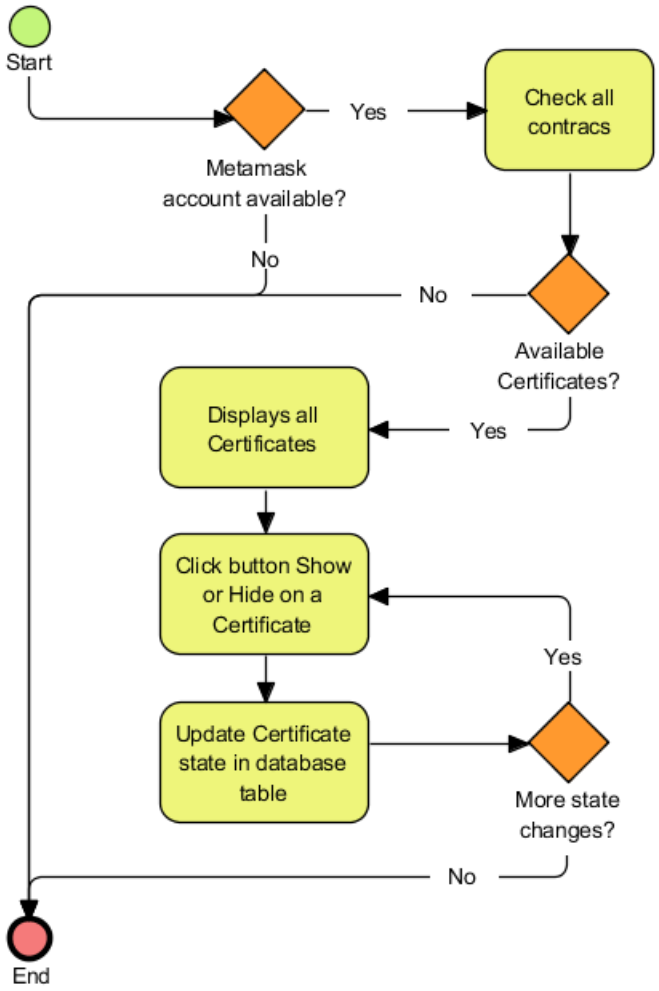


Image 30 - Manipulate Visibility flowchart

The flowchart of the manipulate visibility has the same steps as View Certificates, with the extension of changing the view. The complete steps of are:

1. Necessary to have a Metamask account authentication for the web application automatically assume the verified blockchain account. Otherwise the user cannot change the certificates visibility;
2. The application checks all the available Certificate smart contracts in the Transaction table and displays them as a list. According to the table's Show field the "Show" or "Hide" button appears to change the certificate visibility, in oppose to the previous value[31]
3. If certificates are found in the table, the web application fetches the information on the blockchain for each transaction;
4. After the certificate information is retrieved from the blockchain, the image hash is used to get the image file from the IPFS server;
5. When there are available certificates, the control service displays a certificate list, based on the information retrieved, from every certificate;
6. After the list if complete, the Learner can then change the visibility of the certificate to Show or Hide (based on the appearing buttons), ensuring the update certificate's visibility to the Transaction table, on the Show field.

[31] If Show field is "true" then the "Hide" button appears; if it is "false" then the "Show" button appears.

5.4.3 Certificate Management

Certificate Management is a set of operations that issue smart contracts and call their functions to manipulate its field. Only the verified Educational Institutions accounts[32], can manage these operations.

The following items explains the three certificate management processes Insert Certificate, Modify Certificate and Delete/Reactivate Certificate.

5.4.3.1 Insert Certificate

Represents the issue of the certificate to the blockchain, with the required fields to be filled by the Educational Institution. Operates with the ***constructor*** function, to enable the creation of a new certificate inside the blockchain

The flow of how the deploy of the certificate processes is shown in the Image 31.

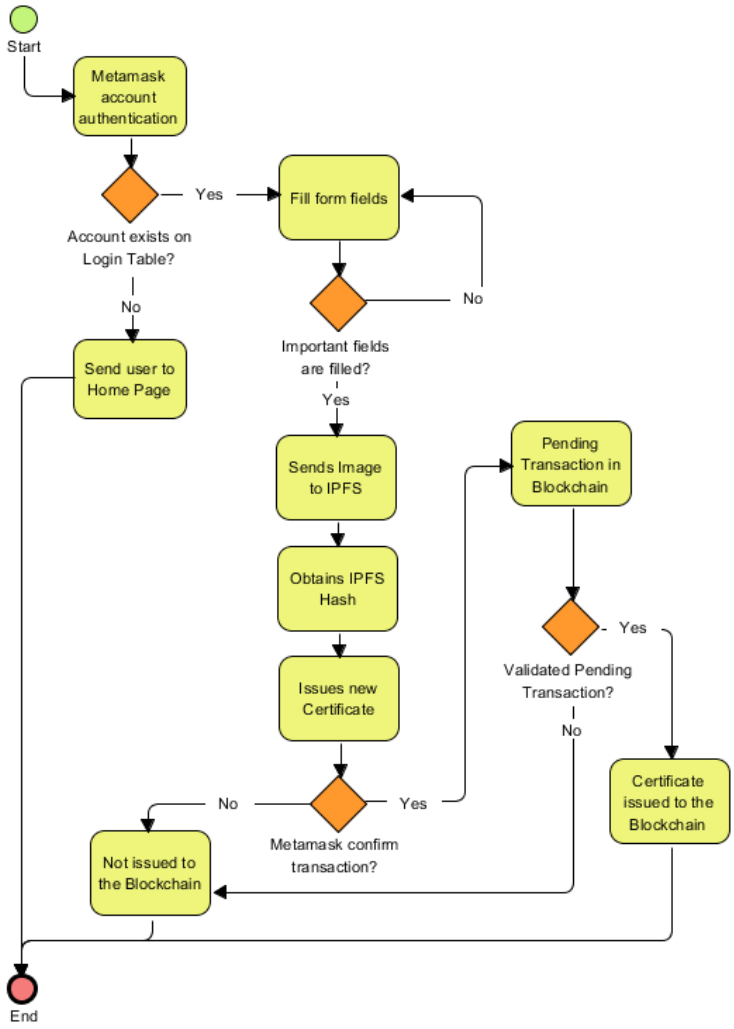


Image 31 - Insert Certificate flowchart

[32] In the Login table and authenticated using the Metamask plugin.

The steps for the deployment are detailed in the following numbering:

1. Metamask account authentication, if not validated redirects user to the Homepage;
2. Requires the information in the web application form (Learner account and image file) to be correctly filled. Shows message if the form hasn't the information filled;
3. After the information in the form is completed and sent, the image is sent to the IFPS server and retrieves a hashed text;
4. The filled information and the hashed text are then allocated to the smart contract, by calling the *constructor* method, and it is sent by the Educational Institution blockchain account;
5. Metamask plugin asks if the user confirms the issue of the smart contract, this step also states the price of the smart contract to the blockchain. If confirmed the issued smart contract is allocated inside the blockchain, as a pending transaction. It is important that the account has enough ether to confirm the message, otherwise it cannot deploy the transaction;
6. When the pending transaction is validated, through the blockchain mining, the transaction is allocated to a block, where can be extracted its information. If not validated through the blockchain mining, the pending transaction is removed.

5.4.3.2 Modify Certificate

An operation that manipulates the issued Certificate smart contract, to modify the available variables. Operates with the function ***setContract*** in the issued Certificate smart contract, this function allows to change the value of the variables *sendToAccount*, *ipfsHash* and *description*.

A smart contract cannot be directly modified, only by calling a function that allows the smart contract to modify the internal variables, these functions are then issued to the blockchain as new transactions.

The flow representing the operation of this operation is demonstrated in the image Image 32.

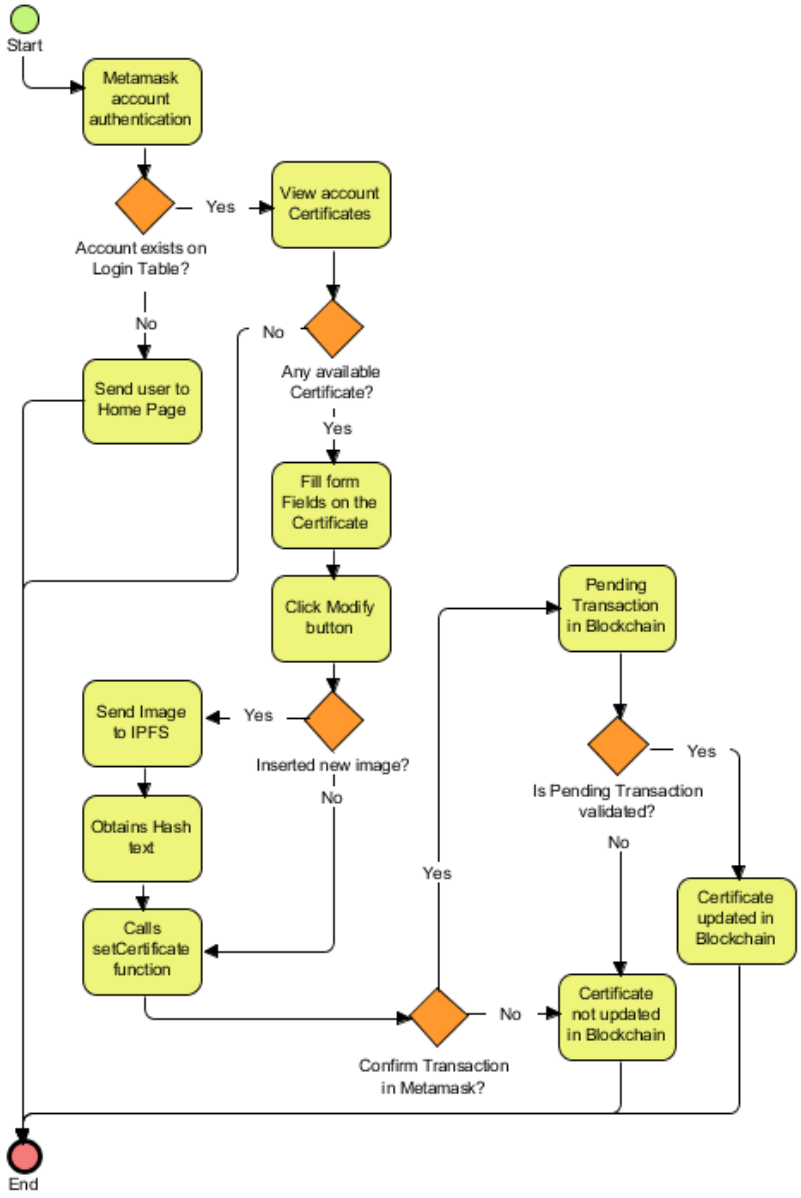


Image 32 - Modify Certificate Flowchart

The flowchart has the following steps:

1. Metamask account authentication, if not validated redirects user to the Homepage;
2. Fill the necessary form fields in of one certificate on the available Certificates list and click the Modify button to send the modification, with a call to the *setContract* on the chosen Certificate smart contract;
3. If the IPFS image is changed, the image is sent to the IPFS server and then retrieves a hash text. This hash text is added to the *setContract* with the other filled information on the form;
4. The call to the function on the previous smart contract is then deployed by the Educational Institution, confirming the Metamask message and requiring a small amount of ether to implement the change. If confirmed the issued smart contract is allocated inside the blockchain, as a pending transaction;
5. When the pending transaction is validated, through the blockchain mining, the transaction is allocated to a block, where can be extracted its information. If not validated through the blockchain mining, the pending transaction is removed.

5.4.3.3 Delete/Reactivate Certificate

The Delete/Reactivate Certificate allows the Educational Institution to inactive the issued smart contract, making that the web application does not list the smart contract in the application, or to reactivate the inactive certificate, to display the certificate. This calls a function on the Certificate smart contract that is ***setActive***, manipulating the active variable.

As stated in previous chapters, is not possible to delete a certificate, since all the information inside the blockchain is immutable once validated and inserted. It is only possible to change the values by calling the functions inside the smart contract, however it requires a small amount of ether to implement the change.

The usage of the Delete description is to make clear for the end user to comprehend the operation definition, as the transaction of the smart contract isn't ever deleted from the blockchain.

Delete/Reactivate Certificate functions similarly as the Modify Certificate. The flow is represented in the Image 33.

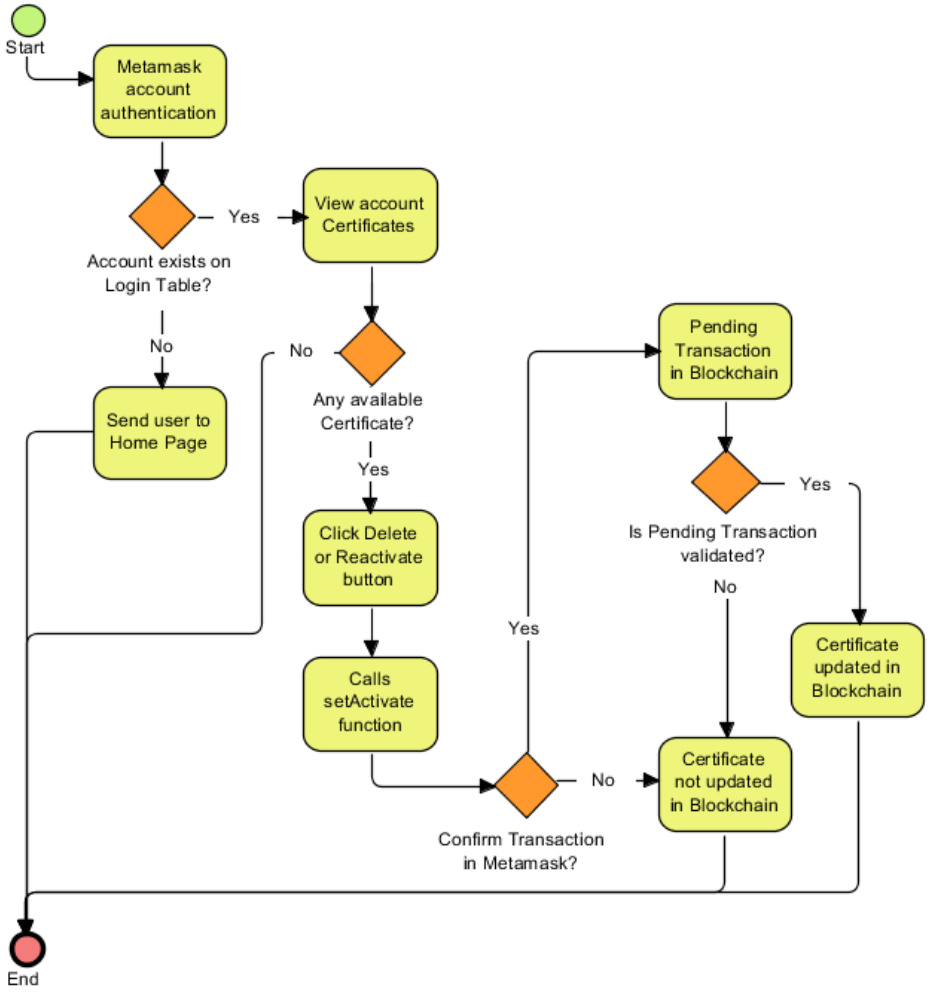


Image 33 - Delete/Reactivate Certificate flowchart

The flowchart has the following steps:

1. Metamask account authentication, if not validated redirects user to the Homepage;
2. Change a specific Certificate smart contract active value by the clicking on the “Delete” or “Reactivate” button. The active argument is allocated as “false” or “true” based on the clicked button, respectably;
3. A call to the function *setActive* on the selected smart contract is deployed by the Educational Institution, after confirming the Metamask message and requiring a small amount of ether to implement the change. If confirmed the issued smart contract is allocated inside the blockchain, as a pending transaction;
4. When the pending transaction is validated, through the blockchain mining, the transaction is allocated to a block, where the change to the active variable has been made. If not validated through the blockchain mining, the pending transaction is removed.

6 Tests and Experimentations

Valuable tests and experimentations are documented in this chapter to obtain a knowledge based on obtained results. Knowledge extracted from the results is important to determine if the project's solution is viable to apply in a real environment, the main blockchain network. Each detailed test is executed three times in the experiments and explained in the result evaluation.

6.1 Tests Details

Tests are conducted on the Ropsten blockchain testing network, issuing the Certificate smart contract, with the Remix Solidity Integrated Development Environment (IDE)[33] and Metamask plugin assistance.

The creation of a new Certificate smart contract to the blockchain is evoked by the *constructor* function, with three arguments: *_sendToAccount*, *_ipfsHash* and *_description*. Of the three arguments the *_description* argument value is variable; the other two arguments have a fixed value of 46 characters, for the *_ipfsHash,* and 42 characters, for the *_sendToAccount*.

The fixed value of 46 characters for the *_ipfsHash* is related to the IPFS's image hash of 46 characters and the *_sendToAccount* is associated to the Ethereum account's hash of 42 characters. Therefore, only the *_description* argument has a manageable character length to the certificate smart contract creation.

[33] Remix Solidity IDE is an open source tool enables the user to write Solidity contracts straight from the browser. Available on the web page: https://remix.ethereum.org .

After the Certificate smart contract issue, the Metamask transaction confirmation has two variables that can be modified to define the fee cost of the transaction, that also influences the validation time. These variables are: Gas Price and Gas Limit, represented in Gwei[34].

So, the tests are guaranteed by the adjustable values on the 3 variables: Description Character Length, Gas Price and Gas Limit. The conducted tests[35] are defined in the Table 5 to be later executed five times in the

Table 5 - Insert Certificate Tests

Test	Description Character Length	Gas Price	Gas Limit[36]
A	0	0,1	760000
B	0	1	760000
C	0	5	760000
D	100	0,1	861500
E	100	1	861500
F	100	5	861500
G	250	0,1	952000
H	250	1	952000
I	250	5	952000

[34] Gwei is a commonly used denomination of ether and represents 0,000000001 ether.
[35] These tests are not based on any standard, they are merely an assumption to possible values on the adjustable variables.
[36] "Gas Limit" values are a rounded value shown in Metamask confirmation transaction, when issued a contract. Also, these values are associate to the "Description Character Length" (only adjustable variable on the smart contract), changing the value of "Gas Limit" as the "Description Character Length" changes.

Table 6 shows the description of the tests conducted for the modify functions present in the Certificate smart contract. These statements are the *setCertificate* and *setActive*.

The arguments for the *setCertificate* function are the *_ipfsHash* (fixed character length of 46) and the *_description* (variable character length); and for the *setActive* function the argument is *_active* ("true" or "false").

Since the character length of *_description* is adjustable, this variable is declared in the Argument column as the character length, for the *setCertificate* function; and the *_active,* for the *setActive* function, with a value of "true" or "false".

Table 6 - Modify Certificate Tests

Test	Method	Argument	Gas Price	Gas Limit
A	***setCertificate***	0	0,1	68000
B	***setCertificate***	0	1	68000
C	***setCertificate***	0	10	68000
D	***setCertificate***	100	0,1	150000
E	***setCertificate***	100	1	150000
F	***setCertificate***	100	10	150000
G	***setActive***	"true" or "false"	0,1	33500
H	***setActive***	"true" or "false"	1	33500
I	***setActive***	"true" or "false"	10	33500

For the detailed tests in the Table 5 and Table 6, the represented value of the Gas Price of "0,1", "1" and "5" demonstrate how slow or fast the transactions are validated, if the value is low then the transaction is not quickly taken by the Miners for validation.

The minimal value represented in the Gas Price is "0,1" and has ten times lower fee cost than the normal Gas Price (of "1"); the value "5" is five times higher fee cost than normal. These selected values shall guarantee a larger discrepancy on the validation time and price cost for each transaction.

6.2 Experiments

The experimentations are executed three times for each test as a transaction to the blockchain, to obtain an estimated validation time and price cost of the transaction.

As previously stated in this chapter, these experimentations are conducted based on the tests of the insert certificate (Table 5), and the modify certificate[37] (Table 6),Table 8 in the Ropsten blockchain testing network, using the Remix IDE to issue and/or change the certificates.

Therefore, the experimentations are shown in the Table 7 and Table 8, respectably, according to three Runs (executions) on different timestamps. This ensures different results on the price cost and validation time on different days. The timestamps, in the Coordinated Universal Time (UTC), are:

- **1st Run** on 9th October 2018, between 9:06pm and 10:07pm (+UTC)
- **2nd Run** on 10th October 2018, between 9:09pm and 10:08pm (+UTC)
- **3rd Run** on 11th October 2018, between 9:08pm and 10:08pm (+UTC)

After these experiments the median value is calculated for each test column, in Table 9, with the following equation:

$$Median\ value = \frac{1^{st}\ Run\ value + 2^{st}\ Run\ value + 3^{rd}\ \ Run\ value}{3}$$

These median values provide better estimation for the analysis in the Results Evaluation section.

[37] Modify certificate includes the Modify Certificate and the Delete/Reactivate Certificate functions

Table 7 - Create Certificate Experimentations

Test	Run	Validation Time (Seconds)	Price Cost (Ether)	Price Cost (Euro)	Transaction Hash[38]
A	1st	67,32	0,000076	0,02	0x548cfc09cfff6164dbf34d19bcac8deaf58688a28a71cd8862aceb109cac1277
	2nd	158,93	0,000076	0,01	0xf41d8427188c216801296e2885949cd75ec5533b5a0c3212e1d4e861136e55d4
	3rd	1841,18	0,000076	0,01	0xd911277a958a7e30ede8d5b8540841f11f9711cb3ec143f6f868125d270ce90e
B	1st	56,48	0,000759	0,15	0x1dcaab1fe05ba27b3d31e2d1d97e668268d98aa18a6138ea38ad9ca74b5ce929
	2nd	81,15	0,000759	0,15	0x938b9d05cd31a26d39855aac90810a709f792030b8a221379d8d96117f4ef37d
	3rd	26,60	0,000759	0,13	0x10b9982608ba72cfc57ce539a3ae5e67bae142388c971e52c6d980e27c73ce1c
C	1st	4,66	0,003795	0,75	0xde43fa9c5f63c40d1686255ca28426dade31f556ce3fd6d4d22669e2b0a1e6a4
	2nd	28,11	0,003795	0,74	0xbb016ec27457536f73f3a295f61165f00f1a0891981b80d0debd2f8af92823d2
	3rd	15,99	0,003795	0,63	0x11e65ec6dd33127e0e4af1d94b4cb4eeee0b1715b66180562c47399980971ae5
D	1st	63,17	0,000086	0,02	0x0aae01eed314d8626b32d0e01fea1590401e40d899a385c3765cb433225b1673
	2nd	257,33	0,000086	0,02	0x9c6d3e0f3909c816d9342da90fc17b3c5a06287fd23ec7782fb1cafe150bae9f
	3rd	48,71	0,000086	0,01	0xee3354ef9a0f9e37a6022fda7cd56b40ec2d8021a8b7eb9ef35b868c08e1d5fd
E	1st	22,45	0,000861	0,17	0xa0bb7c248dd99eff18fb41bf0b266d63c2f057b6365594dd6bcb38351c853f24
	2nd	38,16	0,000861	0,17	0x9d85fe5079d7749cea8856c6a20da7969e469036f1dee858ce773b9c62b4c609
	3rd	18,99	0,000861	0,14	0x07380f677d27d00ff10ae555b3f2b14e798dbacbe1986a5caca6b42c0bb437ce
F	1st	25,06	0,004307	0,85	0x28ddb0b20949f90c55e6792f182b1a94e5ed4b581608122a80f3df9f84259d34
	2nd	7,71	0,004307	0,84	0x8010e5e32eb84b74c3dde92d7e0aceee1c6b2263d86d3b22ccc17d14ad67f641
	3rd	15,09	0.004307	0,72	0x51933f0836aebfb6d4c51ae5085af001ce9eee209efd71d616386f6fd08a5128
G	1st	26,77	0,000095	0,02	0xd8011892fbddc87f3711e6e3a2ab3f124e0f74695d4f59ffd13def39eb7647b9
	2nd	74.93	0,000095	0,02	0x17880c46511e1caf233410f521f236e66c1fa0300b19639023fca22f0a9c92c7
	3rd	22,17	0,000095	0,02	0xeaf85148bbe31a48884bb576f63743cc17f300e4f7e82bce4a9e11644b1f675d
H	1st	8,14	0,000952	0,19	0x450cb5b47d1c1c6dba517aaef75842e168ef9143d7be97062ac4973fcc07d8ef
	2nd	42,99	0,000952	0,19	0xe3ca7d870678597e50772a9a40685f2d561ded1b1e0ccfaae71735ce9bd45ef3
	3rd	41,68	0,000952	0,16	0xe48724411e19f6c6fdf579f7cd1a4f997cb62d2ed1277f488092097ff61cef1a
I	1st	31,08	0,004759	0,94	0x8694825611e0296b1981e43606197a0b13dd7226f4cbd59cccb7b9f30a06cdcd
	2nd	6,76	0,004759	0,93	0x2ac00d14247e2fca66eea3b7997def1c44f2aadded75d3971befbcb5775a8568
	3rd	28,66	0,004759	0,79	0x1f4bfa0dda450f601b47a2327ccd5c1bc9506f03963d9daa4f5d54bfd5176591

[38] Transaction hash on the Ropsten blockchain testing network. If information persists on the testing network, more information can be checked using the web page Ropsten Etherscan: https://ropsten.etherscan.io .

Table 8 - Modify Certificate Experimentations

Test (Method)	Run	Validation Time (Seconds)	Price Cost (Ether)	Price Cost (Euro)	Transaction Hash[39]
A (*setCertificate*)	**1st**	12,16	0,000003	0,00	0xe6cc6b9b6d2b8cdb806c24aa4cede3bcc3d97175ae2a1eb8efee4278eef08917
	2nd	48,19	0,000003	0,00	0x9f942e10cf35ae2482a0e8d4def1455b7ef611d24db760ce07d3846d32fdf702
	3rd	27,74	0,00003	0,00	0xa3cc3c62d0a9ec18a3979d3b0e933438b0b230b161a0a053a1d14b10800d2094
B (*setCertificate*)	**1st**	68,62	0,000034	0,01	0x15650767c95b764b652c9d3220dc1ae55c0d74d65e58a964e31b60bf57e77b57
	2nd	37,72	0,000034	0,01	0x38f7269cd49654caa3ad0e8cc472270ce41217d9cfb0639aec2f5f0a4b180b1f
	3rd	16,12	0,000034	0,01	0x40ed0269c965341fcfa314d1a72e1bdd5f59e932466073db0acb427f138d9bcf
C (*setCertificate*)	**1st**	12,56	0,000169	0,03	0xe555914bf5d0290d22018d6944bf6c2c760c1a3424409da65404688eef40c237
	2nd	9,52	0.000169	0,03	0x8aed79150c1e7d8f94473d83325316ef7fb8fe38c908afc14552564682d4b1a5
	3rd	18,20	0,000169	0,03	0x12127391e617ae0217496555be68a489af22ed2dcf2b3bd2acd47139522998bf
D (*setCertificate*)	**1st**	130,01	0,000015	0,00	0x280180379a76b41b614bc4a668492a3ebfa7b937fc1cc3828f441d60af3106d2
	2nd	51,04	0,000015	0,00	0x0eb71ee32e860880eced9966769c1b34e9e8a1554215b9f3c98dd8e03730628a
	3rd	35,56	0,000015	0,00	0x40d6f116dddeb593c114a90ed734da189575f41a74189024a14ecd0eb0535dc1
E (*setCertificate*)	**1st**	27,25	0,000150	0,03	0x1b823509b0c2aa601854b9ea6f536c8c2f575ed173c5683f925b97259d9dded3
	2nd	30,89	0,000150	0,03	0xc96e6b35daecc0b556340b38219b526655cd25ca7ee36f50cbc3ccc1f64bdac5
	3rd	11,91	0,000150	0,02	0x5ec2f0e7bb29cb1fb398ac58d17d63367f5786cdca39e2626a984afe08c1eec7
F (*setCertificate*)	**1st**	20,50	0,000748	0,15	0xde3e1fdc5a72b0f2d4b2573730eff9a7c57ef3d91bebae8c482bdf326fa5462c
	2nd	17,99	0,000748	0,15	0xd411ea8ac7575b993c971018195487a24c134d408448cf2d9d737177e82729f2
	3rd	9,35	0,000748	0,13	0x3044d3ecf1b23bd0ed2b54084fccb2ef0a5284c662cf97248e2b3f82fb3cc51e
G (*setActive*)	**1st**	49,72	0,000003	0,00	0xc995c9567a9fd8b248e05d8a12a984702a7f8fa77bd39bba614f562f758b1450
	2nd	662,60	0,000003	0,00	0xdc47218441a1cb41e9767085e979f622e38f7ef9c0cfdac809126c3c044c23e2
	3rd	11,10	0,000003	0,00	0x930253707bd0417913670a8ba2e4dac2e2d03400245ec4a8a10b0ade8273fb65
H (*setActive*)	**1st**	7,86	0,000033	0,01	0x9f351abfe892b5dda774f7c63a7b3fad59de93d778cb2ba07329afc869be5bfd
	2nd	110,55	0,000033	0,01	0x02020ec2ab41c81acb9b37046cd2b65b883fe94161fdc91d297d360a2e34abde
	3rd	10,01	0,000033	0,01	0x6c168c7ba5a729b1c888301594925a17a0e74f13a6dd2c96178deb8737bb1f88
I (*setActive*)	**1st**	13,27	0,000167	0,03	0xa96b852f8fb5f2b2e296458bfcc750a0027f73bb936c24d5f0b47c10caddbab5
	2nd	63,26	0,000167	0,03	0x72d8463ae05c34285faf2846e563b5e459a8d84d804654f7c8025d760476b17b
	3rd	8,48	0,000167	0,03	0x3ab2855e05dfd4f09f538d030113e20e70113eabc1de05f5107c8cb4061a1085

[39] Transaction hash on the Ropsten blockchain testing network. If information persists on the testing network, more information can be checked using the web page Ropsten Etherscan: https://ropsten.etherscan.io .

Table 9 - Experimentations Median Calculation

Test	Method	Validation Time (Seconds)	Price Cost (Ether)	Price Cost (Euro)
A	**Insert (constructor)**	689,14	0,000076	0,01
B	**Insert (constructor)**	54,74	0,000759	0,14
C	**Insert (constructor)**	16,25	0,003795	0,71
D	**Insert (constructor)**	123,07	0,000086	0,02
E	**Insert (constructor)**	26,53	0,000861	0,16
F	**Insert (constructor)**	15,95	0,004307	0,80
G	**Insert (constructor)**	41,29	0,000095	0,02
H	**Insert (constructor)**	30,94	0,000952	0,18
I	**Insert (constructor)**	22,17	0,004759	0,89
A	**Modify (*setCertificate*)**	40,89	0,000003	0,00
B	**Modify (*setCertificate*)**	40,82	0,000034	0,01
C	**Modify (*setCertificate*)**	13.43	0,000169	0,03
D	**Modify (*setCertificate*)**	72,20	0,000015	0,00
E	**Modify (*setCertificate*)**	23,35	0,000150	0,03
F	**Modify (*setCertificate*)**	15,95	0,000748	0,14
G	**Modify (*setActive*)**	241,14	0,000003	0,00
H	**Modify (*setActive*)**	42,81	0,000033	0,01
I	**Modify (*setActive*)**	28,34	0,000167	0,03

6.3 Results Evaluation

According to the conducted tests and experimentations, the validation time and price costs in the prototype development have proven to show a speculation on the behavior that the blockchain functions have, according to certain inserted arguments.

Through Table 9, in the median of the Run values, it is possible to assume that the much higher the transaction value in the Gas Price the lower the validation time, however the price value increases.

Since the Runs were issue in different timestamps, is proven to determine different fee price costs in Euros from some transactions, even if not considerably notable in some cases. This is due to the market's Ether price converted to Euros[40] constantly being updated based on the current price market.

Ether price cost always maintains the same on the three runs for every test, yet the transaction time was proven to have given inconstant values. The cause of this inconstancy might be the transaction pool[41] was overly saturated, that the Miners couldn't not answer for their validation in a quickly enough; the other cause might be the Miners couldn't compute the block difficulty fast enough, to insert them to the blockchain network.

These tests and experiments can be proven useful to determine the average cost[42] of an transaction, for example the issue of a Certificate can cost 16 cents (Insert Test E) and the modify cost between 1 to 3 cents (Modify Tests H and E), according to Table 9.

Still these values are only an assumption to some possible costs, as the main blockchain network has higher Gas Price costs than the Ropsten testing network. With a minimum of at least 2 to a maximum of 20 Gas Price, stating that the validation time is completed in about 30 to 2 minutes, respectably[43].

Stated these evaluations, the Educational Institutions can surely send a certificate with a fee cost fewer than 1 Euro to the main public blockchain with a validation time of less than 30 minutes using the prototype's developed solution.

Since the public blockchain values are expected to be affordable for every Educational Institution, the prototype can have two suggested types of implementations: in the private blockchain network or in the main blockchain network.

[40] Price conversion from Ether to Euro in the web page: https://www.coingecko.com
[41] Transaction pool is all the pending transactions on the blockchain network.
[42] Average cost with a transaction suggested "Gas Price" equal to 1.
[43] Main blockchain network values are displayed in the web page: https://ethgasstation.info/, in the "Recommended Gas Prices" zone.

7 Conclusion

It was from the rise of the cryptocurrency that blockchain technology began to gain attention in various segments, from financial to governmental, and was precisely from the impulse offered by the market that this technology glimpses a horizon in the new form of applications.

In the processes of issuing, registering and validating documents, blockchain is proven to completely use these processes. Among with the benefits that can be obtained: independence of centralizing entities, immutability of information, irrefutability and transparency of service. All this promoting a change in the society's way to process certificates and the usage simplicity to reach those involved in the process.

The great potential of the blockchain lies in its integration with other forms of technology and devices. In short, this technology can make it much easier with reduced costs and without the need for a centralized management, since all operations in the blockchain are accordingly analyzed before being effectively registered in the network.

7.1 Work Summary

The thesis work was accomplished with the expected outcomes: important topics were covered, value evaluation was articulated, prototype design and implementation were documented, and the experiments for the detailed tests were conducted, with the appropriate result conclusions.

According to explanation on the documented chapters, this project's intentions were successful in acquiring the desirable conclusions. Resulting with the closure that blockchain are an adaptable technology that can assign the educational certificates to their blocks, ensuring that this information is kept verified and immutable in the blockchain.

The choice between a main or a private blockchain network depends mostly on the approach offered by the developed application, as both are appropriate to implement, with the usage of an external server for allocating files (for example IPFS). The downside of the main blockchain network implies some ether cost, yet is proven to have valuable use in its constant availability that the private blockchain networks might not guarantee.

7.2 Limitations and Future Work

The most noticeable limitation is related to the developed prototype, due to the restriction of the tests and experiments for the smart contracts, deployed in the Ropsten testing network. This highlights the testing limitations in obtaining real price costs and real validation times.

Other limitations in the prototype development is associated to not safeguarding any security implementations for the website management and the non-existent creation of server nodes in the private blockchain network and the private IPFS.

These limitations can prove to limit the life-time and recoverability of the project in a real environment. Nonetheless these limitations were intended, since the thesis main consideration was to deploy smart contracts, extract the deploy information from the blockchain network, and study the obtained results.

A possible future work comes to ensuring the correction of the limitations, as well as adjusting the account management, making a lot simpler and more accessible in different devices, similarly approach as BlockCert and TrueRec projects.

As an extra topic regarding the smart contract application in educational areas, a solution can be implemented to create an entertaining and challenging application to quiz the students of a certain theme, rewarding them for the correct answer with the ether.

Each student that wanted to participate must pay a small fee and answer the quiz, in an open text response. All this information was kept confidential until the specified deadline. After the deadline the answers can be viewed by an assigned group of juris and verified for the most corrected response (without knowing which student answered the question). The winner takes the price of the quiz.

The quiz application can be implemented using smart contracts and can be an extension for this project's application, in which can be conjoint to view their certificates and their quiz achievements in a learner's account.

References

A. Castro, 2017. *Blockchain And IoT: A Perfect Match?.* [Online]
Available at: https://blockgeeks.com/blockchain-and-iot-a-perfect-match/
[Accessed 06 January 2018].

A. Grech, A. F. Camillery & A. Inamorat, 2017. *Blockchain in Education.* [Online]
Available at: https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/blockchain-education
[Accessed 02 February 2018].

A. Henver, S. T. March, J. Park & S. Ram, 2014. *Design Science in Information Systems Research.* [Online]
Available at: https://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf
[Accessed 21 September 2018].

A. Hevner, 2015. *MC: Design: Alan Hevner - Robust Processes of Design Science Research Available at: https://www.youtube.com/watch?v=gdCYH_a4hzY*
[Interview] (4 October 2015).

A. Kosba, A. Miller, E. Shi & Z. Wen, 2016. *The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts.* [Online]
Available at: https://eprint.iacr.org/2015/675.pdf
[Accessed 29 January 2018].

A. Lielacher, 2017. *Top 5 Biggest ICOs (by Return on Investment).* [Online]
Available at: https://www.bitcoinmarketjournal.com/biggest-icos-roi/
[Accessed 06 January 2018].

A. Osterwalder, Y. Pigneur, G. Bernarda & A. Smith, 2017. *Value Proposition Design.* [Online]
Available at: http://www.orange.ngo/wp-content/uploads/2017/04/value-proposition-design.pdf
[Accessed 20 February 2018].

A. Pozo, 2017. *Uploading an Image to IPFS.* [Online]
Available at: https://medium.com/@angellopozo/uploading-an-image-to-ipfs-e1f65f039da4
[Accessed 29 June 2018].

A. Rosic, 2017 [a]. *Ethereum Token.* [Online]
Available at: https://blockgeeks.com/guides/ethereum-token/
[Accessed 10 February 2018].

A. Rosic, 2017 [b]. *Smart Contracts: The Blockchain Technology That Will Replace Lawyers.* [Online]

Available at: https://blockgeeks.com/guides/smart-contracts/
[Accessed 06 January 2018].

Alexandre & Andrew, 2016. *So what is this Blockchain all about?.* [Online]
Available at: http://www.honner.com.au/news-and-insights/news/so-what-is-this-blockchain-all-about
[Accessed 21 February 2018].

B. Arvanaghi, 2018. *Explaining the Genesis Block in Ethereum.* [Online]
Available at: https://arvanaghi.com/blog/explaining-the-genesis-block-in-ethereum/
[Accessed 01 July 2018].

Blockgeeks, 2017. *What is Ethereum Metropolis.* [Online]
Available at: https://blockgeeks.com/guides/ethereum-metropolis/
[Accessed 16 August 2018].

Blockstars, 2015. *Understanding Blockchains.* [Online]
Available at: https://www.slideshare.net/BlockstarsIO/understanding-blockchains
[Accessed 24 February 2018].

C. Bovaird, 2017. *Top 5 ICO Crowdfunding Success Stories.* [Online]
Available at: https://www.bitcoinmarketjournal.com/top-ico-success-stories/
[Accessed 06 January 2018].

C. Guiterrez & A. Khizhniak, 2017. *SAP Verifies Academic Credentials Using Blockchain and Cloud Foundry.* [Online]
Available at: https://www.altoros.com/blog/sap-stores-academic-credentials-using-blockchain-and-cloud-foundry/
[Accessed 06 September 2018].

C. Jagers, 2016. *Verifiable Credentials on the Blockchain.* [Online]
Available at: https://medium.com/learning-machine-blog/blockchain-credentials-b4cf5d02bbb7
[Accessed 18 February 2018].

C. Zorzini, 2018. *What Are Gas Limit and Gas Price for Ethereum Transactions?.* [Online]
Available at: https://unblock.net/what-are-gas-limit-and-gas-price/
[Accessed 2 October 2018].

D. P. Franco, F. D. Barboza & N. M. Cardoso, 2013. *A Secure Method for Authenticity Verification of Handwritten Signatures.* [Online]
Available at:
https://pdfs.semanticscholar.org/420d/eb9a1f9b1fd9a65b29c22bbf7b62f689b69b.pdf
[Accessed 21 February 2018].

D. Palmer, 2016. *7 Cool Decentralized Apps Being Built on Ethereum.* [Online]
Available at: https://www.coindesk.com/7-cool-decentralized-apps-built-ethereum/
[Accessed 06 January 2018].

D. Tapscott & A. Tapscott, 2017 [a]. *Realizing the Potential of Blockchain.* [Online]
Available at: http://www3.weforum.org/docs/WEF_Realizing_Potential_Blockchain.pdf
[Accessed 12 February 2018].

D. Tapscott & A. Tapscott, 2017 [b]. *The Blockchain Revolution and Higher Education.* [Online]
Available at: http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?vid=1&sid=dd602532-1db3-444e-849c-41966aa03b46%40sessionmgr120
[Accessed 04 February 2018].

E. Kozliner, 2017. *Merkle Tree Introduction.* [Online]
Available at: https://medium.com/@evankozliner/merkle-tree-introduction-4c44250e2da7
[Accessed 23 February 2018].

E. Mik, 2017. *Smart contracts: terminology, technical limitations.* [Online]
Available at: http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?vid=1&sid=4309b20b-5f75-4f12-bd89-ddea5b49f301%40sessionmgr103
[Accessed 11 February 2018].

E. Pollard, W. Hirsh, M. Williams, J. Buzzeo, R. Marvell, A. Tassinari, C. Bertram & L. Fletcher, 2015. *Understanding Employers' Graduate Recruitment and Selection Practices: Main report.* [Online]
Available at:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/474251/BIS-15-464-employer-graduate-recruitment.pdf
[Accessed 18 February 2018].

Ethereum Community, 2016. *Ether - Ethereum Homestead Documentation.* [Online]
Available at: http://www.ethdocs.org/en/latest/ether.html#what-is-ether
[Accessed 06 January 2018].

Ethereum Comunity, 2014. *White Paper - Ethereum.* [Online]
Available at: https://github.com/Ethereum/wiki/wiki/White-Paper
[Accessed 26 January 2018].

Ethereum Frontier Release, 2015. *The Frontier Release.* [Online]
Available at: https://ethereum.gitbooks.io/frontier-guide/content/frontier.html
[Accessed 16 August 2018].

F. Tschorsch & B. Scheuermann, 2016. *Bitcoin and beyond: A technical survey on decentralized digital currencies.* [Online]
Available at: https://eprint.iacr.org/2015/464.pdf
[Accessed 29 January 2018].

G. Chen, B. Xu, N. Chen & M. Lu, 2018. *Exploring blockchain technology and its potential applications for education.* [Online]
Available at: https://slejournal.springeropen.com/track/pdf/10.1186/s40561-017-0050-x?site=slejournal.springeropen.com
[Accessed 09 February 2018].

G. Gollin, 2008. *Verification of the integrity and legitimacy of academic credential documents.* [Online]
Available at: http://www.hep.uiuc.edu/home/g-gollin/gollin_academic_document_security.pdf
[Accessed 21 February 2018].

G. Wood, 2014. *Ethereum: A Secure Decentralized Generalised Transaction Ledger.* [Online]
Available at: http://gavwood.com/paper.pdf
[Accessed 01 January 2018].

Golang, 2018. *Command Line Options.* [Online]
Available at: https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options
[Accessed 01 July 2018].

J. Ray, 2018. *Releases · Ethereum.* [Online]
Available at: https://github.com/ethereum/wiki/wiki/Releases
[Accessed 17 August 2018].

J. Southurst, 2016. *Using Blockchain to Fight the Fake Diploma Nightmare.* [Online]
Available at: https://news.bitcoin.com/blockchain-fake-diploma-nightmare/
[Accessed 06 January 2018].

L. Chung & J. P. Leite, 2009. *On Non-Functional Requirements in Software.* [Online]
Available at:
https://pdfs.semanticscholar.org/2d1e/79e057a9111ea6863378ffeca526a4e41c5f.pdf
[Accessed 14 September 2018].

L. M. Singer & P. A. Alexander, 2017. *Reading on Paper and Digitally: What the Past Decades of Empirical Research Reveal,* s.l.: s.n.

L. Thomas, 2016. *Ethereum Blockchain Mechanism (Proof Of Work).* [Online]
Available at: https://i.stack.imgur.com/afWDt.jpg
[Accessed 21 February 2018].

Linklaters, 2017. *What's Smart About Smart Contracts?.* [Online]
Available at: https://www.isda.org/a/6EKDE/smart-contracts-and-distributed-ledger-a-legal-perspective.pdf
[Accessed 15 February 2018].

M. Dejen & H. Sekandary, 2008. *Customer Value.* [Online]
Available at:
http://diuf.unifr.ch/main/is/sites/diuf.unifr.ch.main.is/files/file/seminars/CRM_FS08/Customer_Value(MichaelDejen_HamedSekandary).pdf
[Accessed 21 February 2018].

M. Merz, 2016. *Potential of the blockchain technology in energy trading..* [Online]
Available at: http://www.ponton.de/downloads/mm/Potential-of-the-Blockchain-Technology-in-Energy-Trading_Merz_2016.en.pdf
[Accessed 30 January 2018].

M. Milutinovic, H. Wu, W. He & M. Kanwa, 2016. *Proof of Luck: an Efficient Blockchain Consensus Protocol.* [Online]
Available at: http://delivery.acm.org/10.1145/3010000/3007790/a2-milutinovic.pdf?ip=89.152.174.26&id=3007790&acc=CHORUS&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1518740171_8a1487857800844315691793 08ed8a5a
[Accessed 09 February 2018].

M. Pilkington, 2015. *Blockchain Technology: Principles and Applications.* [Online]
Available at: https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=2662660
[Accessed 06 January 2018].

M. Scherer, 2017. *Performance and Scalability of Blockchain.* [Online]
Available at: https://umu.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf
[Accessed 11 February 2018].

M. Swan, 2015. *Blockchain - Blueprint for a new Economy.* [Online]
Available at:
https://books.google.pt/books?hl=en&lr=&id=RHJmBgAAQBAJ&oi=fnd&pg=PR3&dq=%22blockchain+3.0%22&ots=XQtBIYXWh2&sig=O6UTFmvGbnbq1QqdmxDWZK6uhPQ&redir_esc=y#v=onepage&q=%22blockchain%203.0%22&f=false
[Accessed 25 August 2018].

M. Wöhrer & U. Zdun, 2018. *Design Patterns for Smart Contracts in the Ethereum Ecosystem.* [Online]
Available at: https://eprints.cs.univie.ac.at/5665/1/bare_conf.pdf
[Accessed 03 October 2018].

MIT Institution, 2016. *Blockcerts.* [Online]
Available at: https://www.blockcerts.org/about.html
[Accessed 03 January 2018].

MIT Media Lab Learning Initiative, 2016. *Blockcerts — An Open Infrastructure for Academic Credentials on the Blockchain.* [Online]
Available at: https://medium.com/mit-media-lab/blockcerts-an-open-infrastructure-for-

academic-credentials-on-the-blockchain-899a6b880b2f
[Accessed 18 February 2018].

P. A. Koen, G. M. Ajamian, S. Boyce, A. Clamen, E. Fisher, S. Fountoulakis, A. Johnson, P. Puri & R. Seibert, 2002. *Fuzzy Front End: Effective Methods, Tools, and Techniques.* [Online]
Available at: http://www.stevens-tech.edu/cce/NEW/PDFs/FuzzyFrontEnd_Old.pdf
[Accessed 10 February 2018].

P. Franco, 2014. Understanding Bitcoin: Cryptography, engineering and economics. In: *Understanding Bitcoin: Cryptography, engineering and economics.* s.l.:s.n.

Pluralsight, 2017. *Blockchain Architecture.* [Online]
Available at: https://www.pluralsight.com/guides/blockchain-architecture
[Accessed 06 January 2018].

R. Dore, 1997. *The Diploma Disease: Education, Qualification, and Development.* s.l.:Institute of Education.

R. Hackett, 2016. *Vitalik Buterik - Can this 22-year-old coder out-Bitcoin Bitcoin?.* [Online]
Available at: http://fortune.com/ethereum-blockchain-vitalik-buterin/
[Accessed 08 February 2018].

S. Meiklejohn, M. Pomarole, G. Jordan & K. Le, 2013. *A Fistful of Bitcoins: Characterizing Payments Among Men with No Names.* [Online]
Available at: http://www0.cs.ucl.ac.uk/staff/S.Meiklejohn/files/login13.pdf
[Accessed 19 February 2018].

S. Shin, 2012. *Education.* [Online]
Available at: http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?vid=1&sid=e03118e1-17b9-470c-9c25-1f840da603d7%40sessionmgr103
[Accessed 08 February 2018].

S. Taylor, 2015. *Blockchain: understanding the potential.* [Online]
Available at:
https://www.barclayscorporate.com/content/dam/corppublic/corporate/Documents/insight/blockchain_understanding_the_potential.pdf
[Accessed 31 January 2018].

S. Velu, 2017. *What Are Dapps? The New Decentralized Future.* [Online]
Available at: https://blockgeeks.com/guides/dapps/
[Accessed 04 February 2018].

Solidity/Ethereum Community, 2016. *Solidity.* [Online]
Available at: https://solidity.readthedocs.io/en/develop/
[Accessed 12 February 2018].

T. L. Turocy & B. Stengel, 2001. *Game Theory.* [Online]
Available at: http://www.cdam.lse.ac.uk/Reports/Files/cdam-2001-09.pdf
[Accessed 11 February 2018].

TrueRec, 2017. *TrueRec.* [Online]
Available at: https://truerec.io/
[Accessed 14 August 2018].

Unibright.io, 2017. *Blockchain evolution: from 1.0 to 4.0.* [Online]
Available at: https://medium.com/@UnibrightIO/blockchain-evolution-from-1-0-to-4-0-3fbdbccfc666
[Accessed 25 August 2018].

V. Buterin, 2015. *Olympic: Frontier Pre-Release.* [Online]
Available at: https://blog.ethereum.org/2015/05/09/olympic-frontier-pre-release/
[Accessed 16 August 2018].

V. Buterin, 2016. *Ethereum: Platform Review.* [Online]
Available at:
https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57506f387da24ff6bdecb3c1/1464889147417/Ethereum_Paper.pdf
[Accessed 16 August 2018].

V. Trón & H. Jameson, 2016. *Homestead Release.* [Online]
Available at: https://ethereum-homestead.readthedocs.io/en/latest/introduction/the-homestead-release.html
[Accessed 16 August 2018].

Warwick Manufacturing Group, 2007. *Product Excellence using Six Sigma - Quality Function Deployment.* [Online]
Available at:
https://warwick.ac.uk/fac/sci/wmg/ftmsc/modules/modulelist/peuss/slides/section_6a_qfd_notes.pdf
[Accessed 10 February 2018].

Annex A - Design Science Research Guidelines

Table 10 information was extracted from the “Design Science in Information Systems Research” (A. Henver, et al., 2014), in page 83.

Table 10 - Design Science Research Guidelines

Guideline	Description
Guideline 1: Design as an Artifact	• Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	• The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	• The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	• Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	• Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	• The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	• Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Annex B - Ethereum Yellow Paper Representation

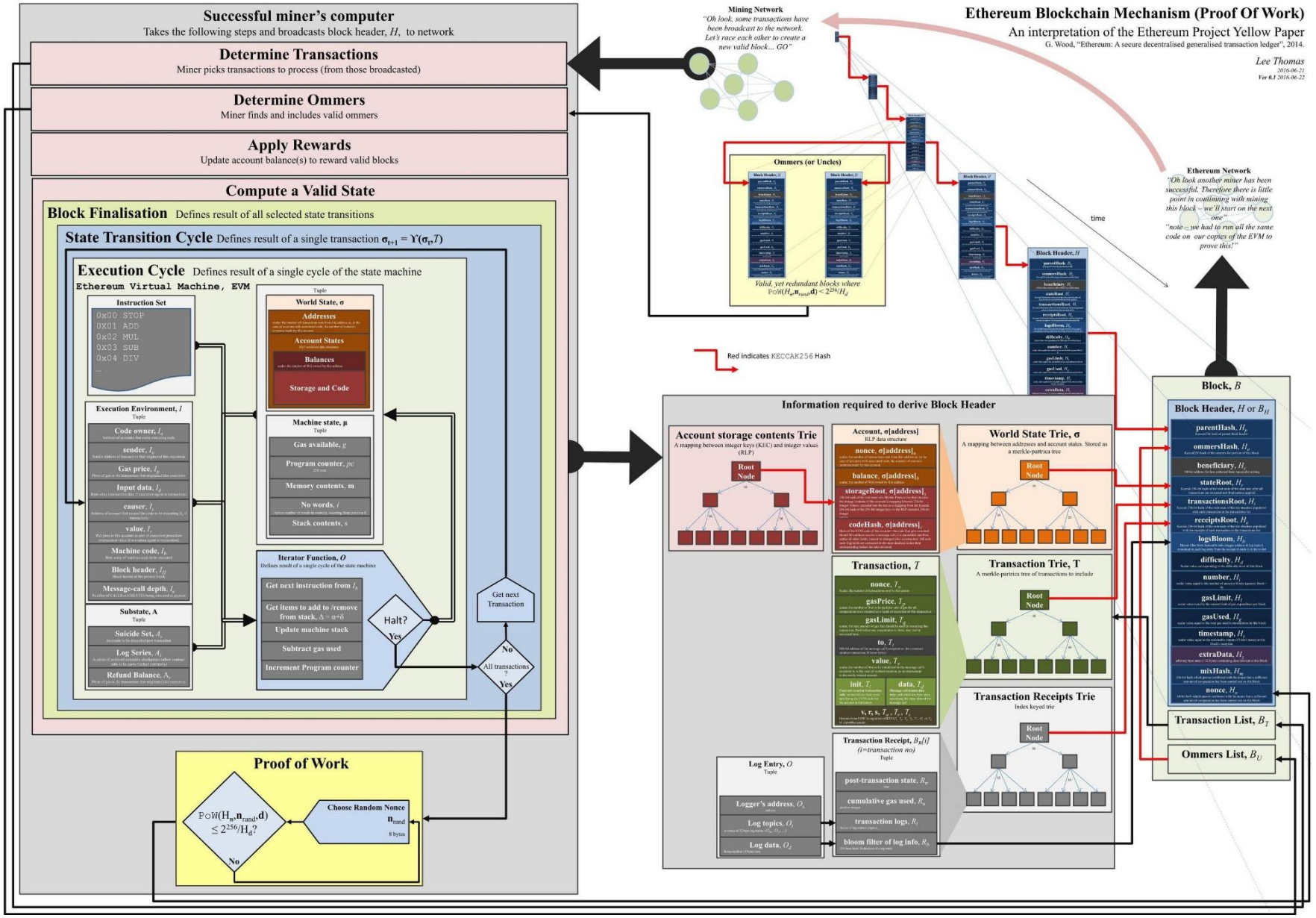


Image 34 - Ethereum Yellow Paper Representation (L. Thomas, 2016)

Annex C - Base 64 Image to Blockchain

To issue an image file to the blockchain the image file must firstly be converted to Base64 encoding[44], since smart contracts code doesn't offer any other form to storage images files, beside *string* text. Converting a certificate image has immense characters that are needed to be stored to the blockchain, to completely ensure the image visibility when the Base 64 is decoded. Image 35 represent the result of character size in a converted image (size 76KB) to Base 64.

Image 35 - Conversion image file to Base 64

Using the simple smart contract of ImageStorage, in Code 4, it is possible to obtain an assumption on the cost of the inserted image to blockchain per 1000 characters stored[45].

```
pragma solidity ^0.4.24;
contract ImageStorage {
    string public imageBase64;

    constructor(string _imageBase64) public {
        imageBase64 = _imageBase64;
    }
}
```

Code 4 - Image Upload smart contract

- Storage of 1000 characters
  (Transaction Hash: 0xce1570d9ec639f47325e9873d3db730a592217c3d5704db75ca6f25405444430)
  - Cost: ≈0,16 Euros (0.000909262 Ether)
- Storage of 2000 characters
  (Transaction Hash: 0xea76c8598e8ee40ee245161bf56fe75ae31771d4880836978061ebd86ca8b9da)
  - Cost: ≈0,27 Euros (0.001599593 Ether)

$$0{,}27€ - 0{,}16€ = 0{,}11€\ (price\ to\ store\ 1000\ characters)$$
$$0{,}11€ * 103 = 11{,}33€\ (price\ to\ store\ image\ with\ 103000\ characters)$$

The approximate price to store an **76KB image to the blockchain is at least 11,33 Euros**. Still certificate's images can have a much higher size, if it is 500KB the price can easily surpass the 60 Euros for one image stored to the blockchain.

[44] Base64 is an encoding that converts binary files to text representations. More information can be seen on the Wikipedia web page: https://en.wikipedia.org/wiki/Base64 .
[45] Remix IDE doesn't allow to issue a smart contract with an extensive string length. It was tested in a 1000 characters length portion, since it is a more manageable size.

Annex D - Software Installation and Set-up

Annex D.1 - Geth

The commands for the standard installation of Geth are represented in Code 5.

```
sudo add-apt-repository -y ppa:ethereum/Ethereum
sudo apt-get update
sudo apt-get install ethereum
```

Code 5 - Geth Installation

Geth's standard installation has a default project, the main disadvantage of this standard project is the high validation time in the private network, executed by miners. Since in the prototype development is desirable to obtain the transactions immediately to the blockchain, for quick testing, it is recommended to adjust the project's code to avoid this problem of waiting for the validation.

The reason of the higher validation time is caused by Difficulty parameter of the transactions that keeps increasing value and this originates a lot of time for the miners to validate a transaction. To resolve this issue of the increasing Difficulty, the project of Geth in the function *CalcDifficulty* was changed so that validations are immediately inserted to the testing blockchain when mined.

After downloading the Geth project[46], the modification of the file located in the folder path "go-ethereum-1.8.11\consensus\ethash\consensus.go" was accomplish with the Code 6 to avoid the validation time, and then the reload the Geth project (Code 7) has to be done to ensure the modifications in the Geth folder.

```
func CalcDifficulty(config *ChainConfig, time, parentTime uint64,
parentNumber, parentDiff *big.Int) *big.Int {
    return big.NewInt(1)
}
```

Code 6 - *CalcDifficulty* modification

```
install go -> apt install golang-go
reload Geth
```

Code 7 - Reload Geth project

[46] Geth project 1.8.11 download in the web page: https://github.com/ethereum/go-ethereum/archive/v1.8.11.tar.gz

Annex D.2 - Blockchain Set-up (Geth)

To initialize a private blockchain with Geth it Is necessary to prepare a folder and allocate the environment variables to the folder, as shown in Code 8. After this point everything in the Geth blockchain is allocated to the specified working folder.

```
cd ~
mkdir ethereum

echo 'export ethereum_home=/home/blockchaintest/ethereum'
>>~/.bash_profile
echo 'export ethereum_home=/home/blockchaintest/ethereum' >>~/.bashrc
```

Code 8 - Setup folder and environment variables

It is essential to prepare the Genesis block, that composes the block 0 of the private blockchain network, enabling to initialize the blockchain network. The Genesis block includes the following parameters (B. Arvanaghi, 2018):

- **config** group with a set of parameters to configure the network
  - **chainID** sets the network ID (main Ethereum blockchain network is 1)
  - **homesteadBlock** configuration the genesis block (0 signifies homestead block)
  - **eip155Block** represents Ethereum Improvement Proposal (EIP), sets the simple replay attack protection to the blockchain
  - **eip158Block** represents EIP, saves space on blockchain dealing with empty accounts
- **difficulty** determines how hard is to mine a block (the speed of mining also depends on the machine performance)
- **gasLimit** maximum number of computations supported by any block on the chain
- **alloc** group that initializes some accounts in the blockchain
  - **account hash** predefined hash account
    - **balance** associates the ether to the account

Add the genesis file with the structure of Code 9 in the json to the specified working folder.

```
{
    "config": {
        "chainId": 88,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "0x4000",
    "gasLimit": "2100000",
    "alloc": {
"6cb6ae60cbde490423056be4d01c4ce4e685af62": { "balance": "400000" } }
    }
}
```

Code 9 - Genesis *json* file

After the Genesis block is added to the folder, the private network with the genesis block can be initialized. Code 10 represents the initiation of the network, followed by the console initialization to manage the network. The parameters used in the code are (Golang, 2018):

- **--networkid** represents the identification id of the network. The ID 1 represents the public main network, other IDs numbers define the private network in which all the nodes must peered to the same ID. It is important to declare the same networkid as the chainId, so no issues can occur with the sent transactions
- **--port** open a port that allows other nodes to connect to the network
- **--nodiscover** preserves the private network to be undiscovered by exterior machines (only manual peer addition can be established)

```
geth --datadir "$ethereum_home/gethfolder" --networkid 88 --port
"35555" --nodiscover init "$ethereum_home/genesis.json"

geth --datadir "$ethereum_home/gethfolder" console 2>console.log --
port "35555"
```

Code 10 - Initialize private network and open console

Following the initialization of the private network, Geth console can manage all the necessary procedures to operate the private network. The following commands have the most important use for the prototype:

- **admin.nodeInfo** obtains the information regarding the node in the network
- **admin.getPeers** reveals all the peers connected to the network
- **eth.blockNumber** obtains the last block number
- **eth.getBlock(x)** gets the information regarding the x block number
- **personal.newAccount** creates a new wallet account in the network, with a password associated to that account
- **personal.listAccounts** lists all available accounts in the network
- **miner.start(x)** starts x threads of miners to validate the blocks
- **miner.stop()** stops all miners

Annex D.3 - IPFS

The IFPS is desirable to install in nodes machines, since all the nodes communicate with each other to prevent unreachability of the content, or can be simply installed in the main machine (server), as one available node. For the project's content, the installation of IFPS is installed in the virtual machine, resulting in that the information is stored in one machine. The installation of IPFS is in the Code 11 section.

```
tar xvfz go-ipfs.tar.gz
cd go-ipfs
./install.sh
```

Code 11 - IPFS installation

A small verification of the IPFS as a file storage can be accomplished with an image upload to the network. This can be accomplished with the HTML code available in the Annex E - IPFS *Javascript* Image Upload (obtained from the website Medium (A. Pozo, 2017)), inserted the code in an index.html file and open the file to the browser. Browse to the image and upload it to the IPFS network to obtain the hash Image 36.

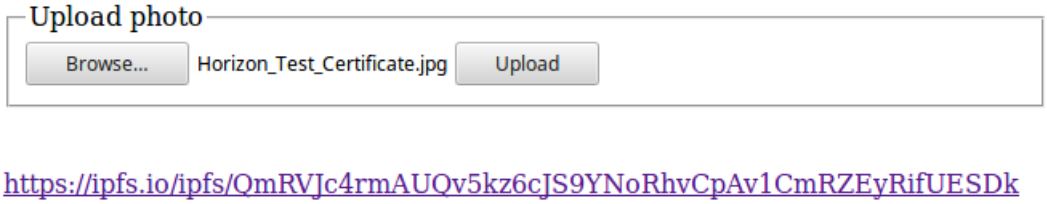


Image 36 - IPFS Image import (with IPFS hash)

Annex D.4 - MySQL

For the database management and administration, it is recommended to install phpMyAdmin, using the following code. The two tools are installed using the Code 12.

```
sudo apt update && sudo apt install mysql-server
sudo apt update && sudo apt install phpmyadmin
```

Code 12 - MySQL and phpMyAdmin installation

After the installation completed, the access to the web interface in the browser is the default URL: http://192.168.1.10/phpmyadmin.

Annex E - IPFS *Javascript* Image Upload

```
<html>
  <head>
    <title>JavaScript file upload</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <script src="https://wzrd.in/standalone/buffer"></script>
    <script src="https://unpkg.com/ipfs-api@9.0.0/dist/index.js"
    integrity="sha384-5bXRcW9kyxxnSMbOoHzraqa7Z0PQWIao+cgeg327zit1hz5LZCEbIMx/LWKPReuB"
    crossorigin="anonymous"></script>
  </head>
  <script type="text/javascript">
    function upload() {
      const reader = new FileReader();
      reader.onloadend = function() {
        const ipfs = window.IpfsApi('localhost', 5001) // Connect to IPFS
        const buf = buffer.Buffer(reader.result) // Convert data into buffer
        ipfs.files.add(buf, (err, result) => { // Upload buffer to IPFS
          if(err) {
            console.error(err)
            return
          }
          let url = `https://ipfs.io/ipfs/${result[0].hash}`
          console.log(`Url --> ${url}`)
          document.getElementById("url").innerHTML= url
          document.getElementById("url").href= url
          document.getElementById("output").src = url
        })
      }
      const photo = document.getElementById("photo");
      reader.readAsArrayBuffer(photo.files[0]); // Read Provided File
    }
  </script>
  <body>
    <form action="/">
      <fieldset>
        <legend>Upload photo</legend>
        <input type="file" name="photo" id="photo">
        <button type="button" onclick="upload()">Upload</button>
      </fieldset>
    </form>
    </br>
    </br>
    <a id="url"></a>
    </br>
    </br>
    <img id="output">
  </body>
</html>
```

Code 13 - IPFS Image Import (HTML)

Annex F - Prototype Demonstration

Annex F.1 – Prototype Homepage

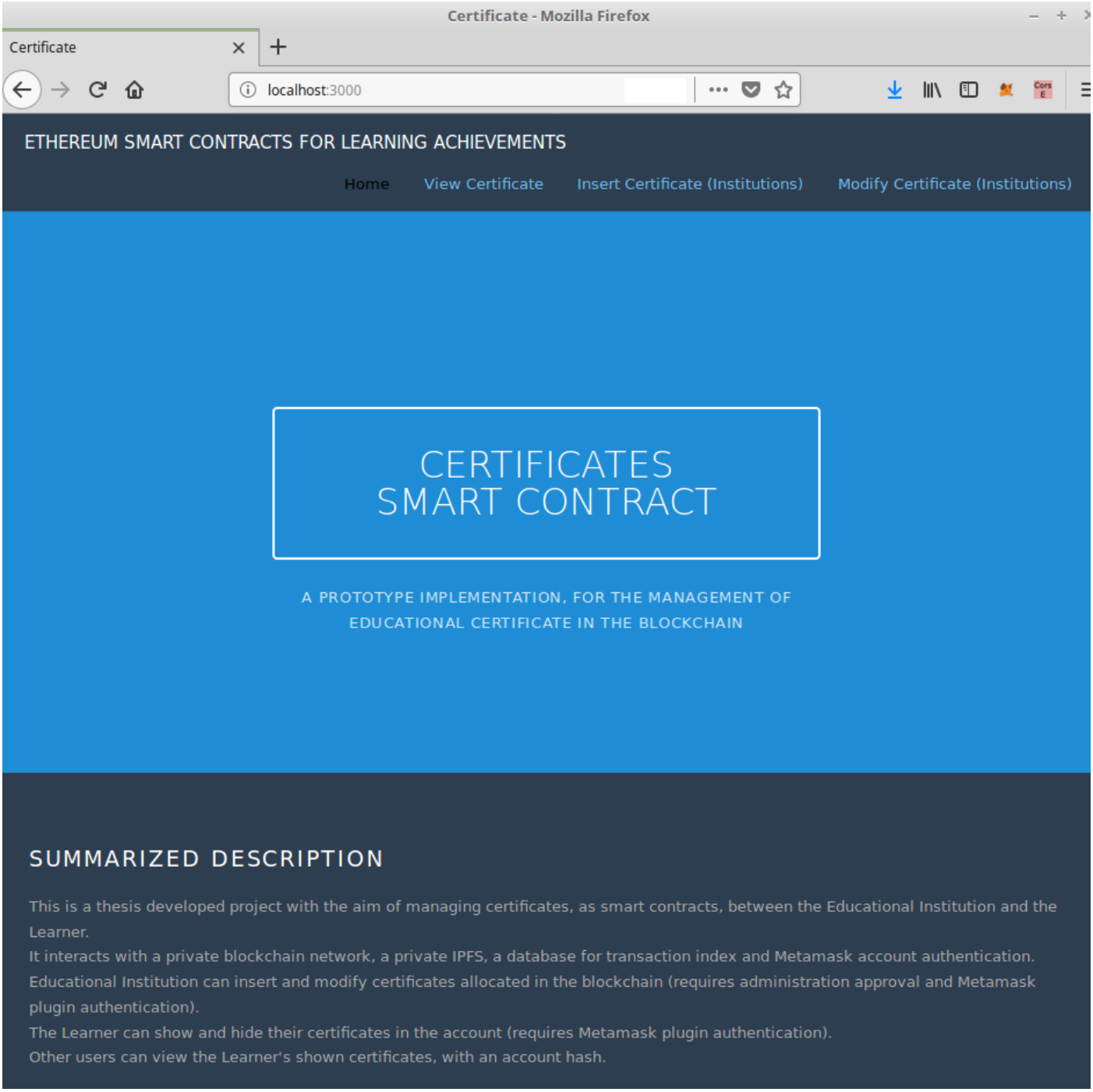


Image 37 - Prototype Homepage

Annex F.2 – Insert Certificate Demonstration

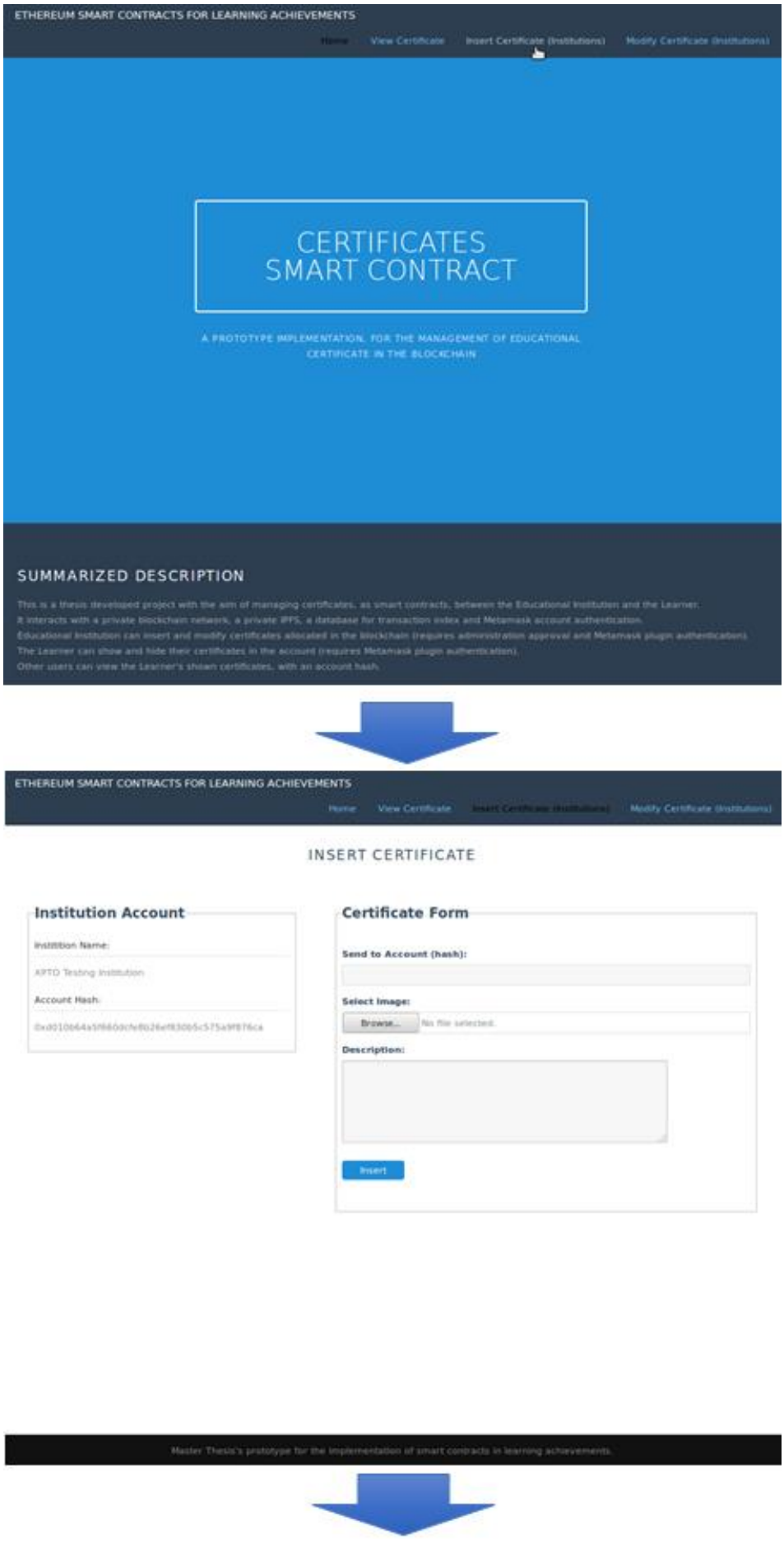


Image 38 - Prototype Insert Certificate (part 1)

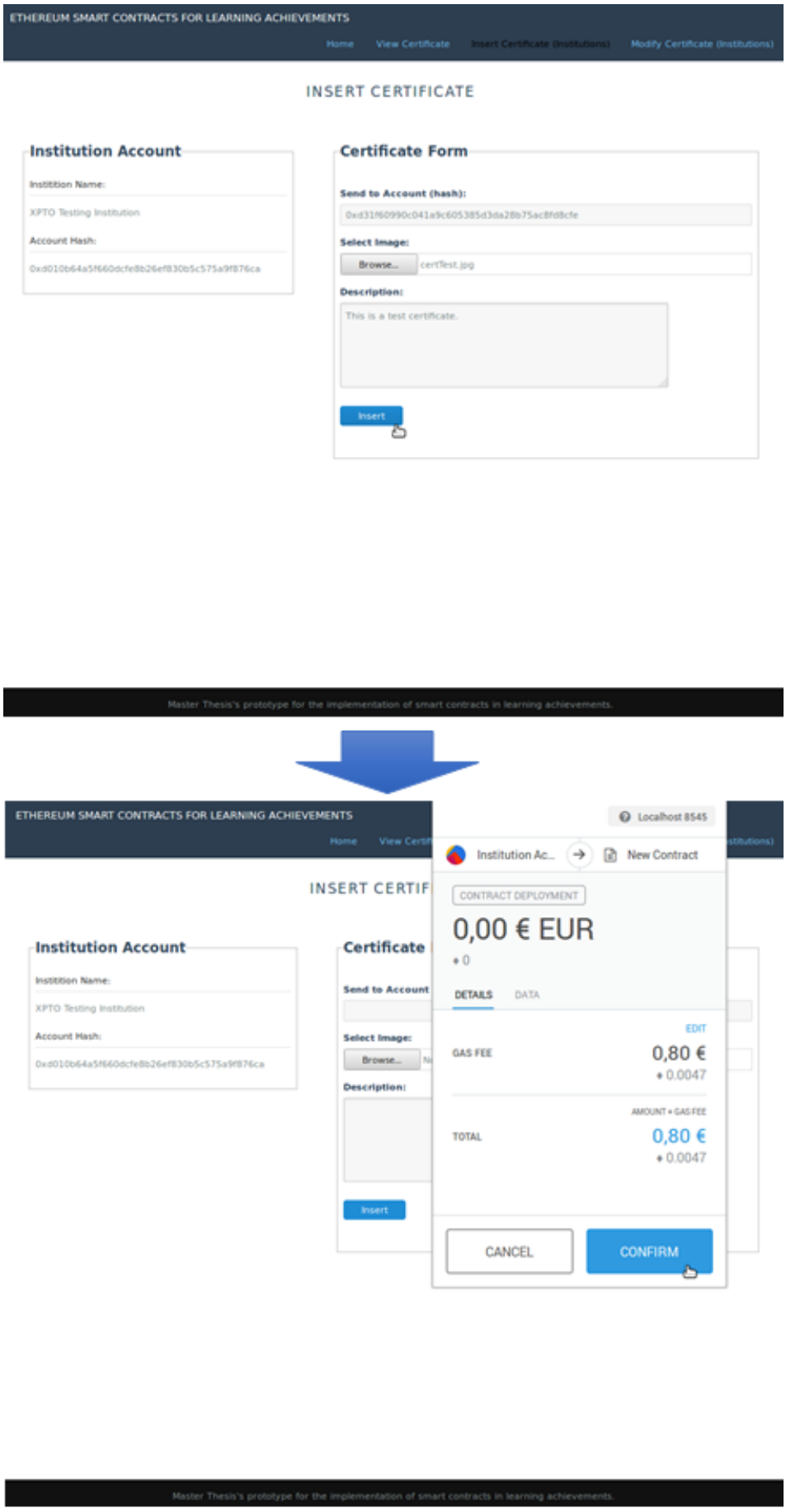


Image 39 - Prototype Insert Certificate (part 2)

Annex F.3 – Modify Certificate Demonstration

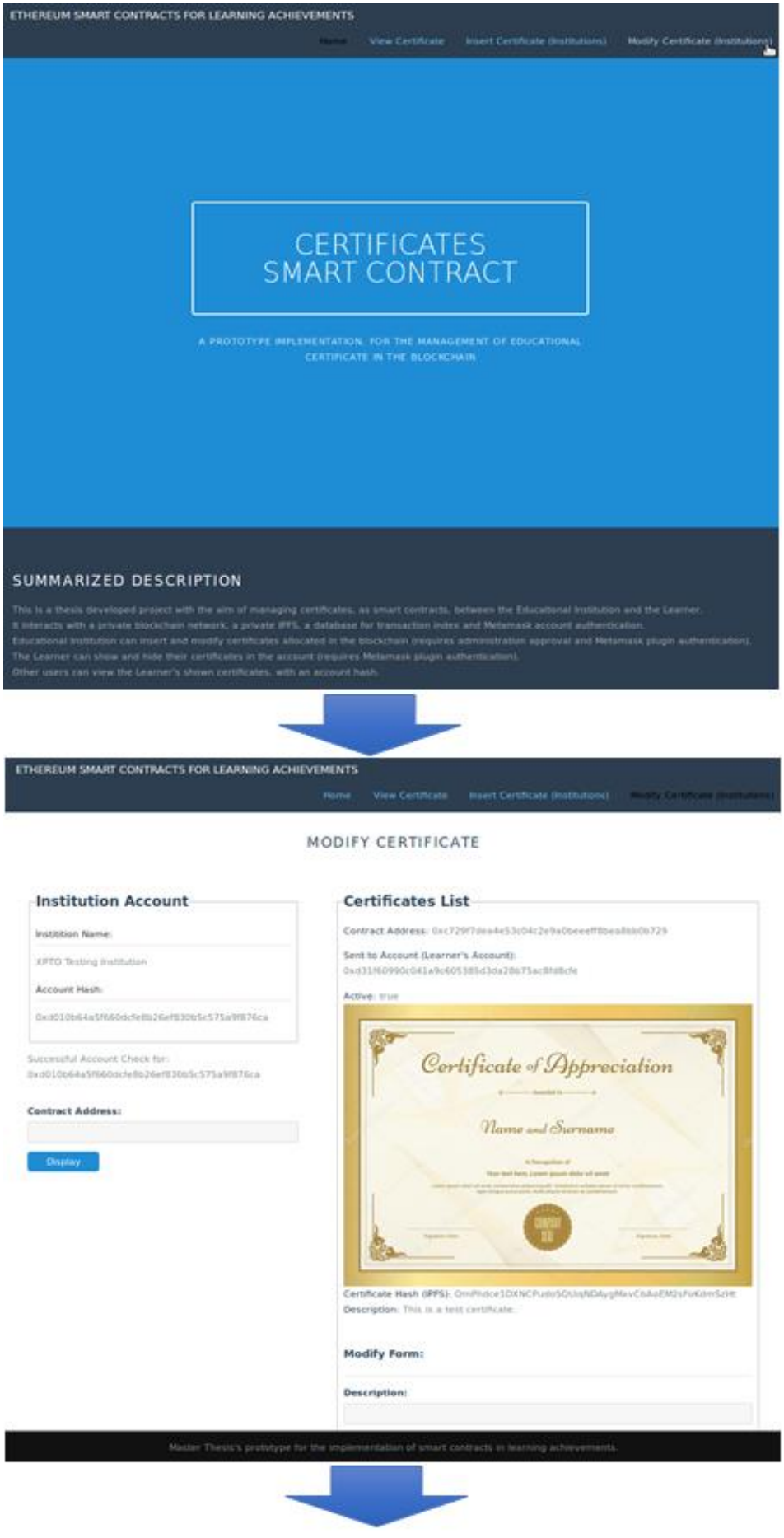


Image 40 - Prototype Modify Certificate (part 1)

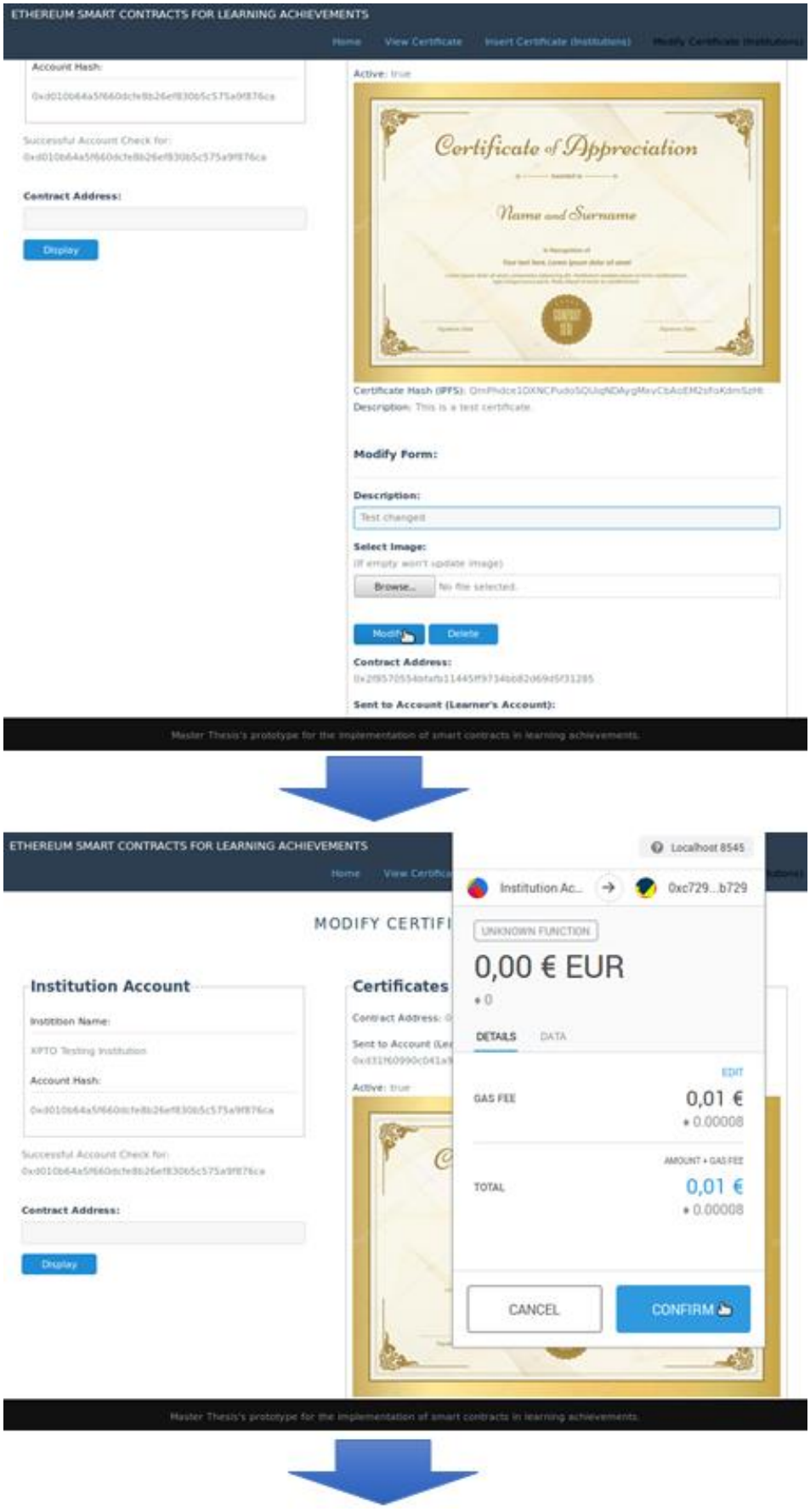


Image 41 - Prototype Modify Certificate (part 2)

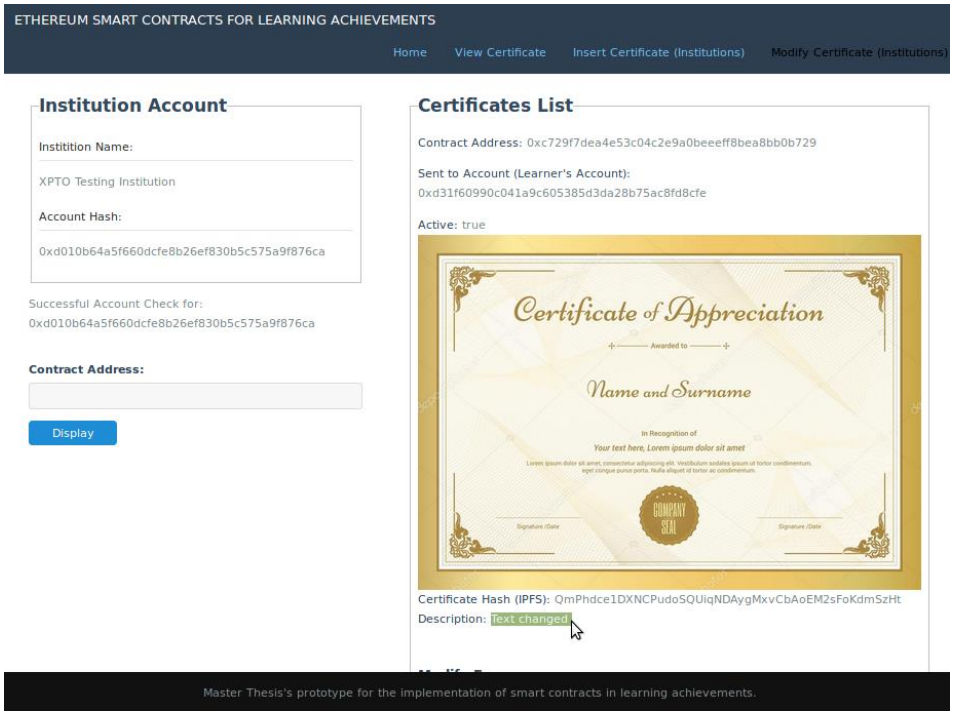


Image 42 - Prototype Modify Certificate (part 3)

Annex F.4 – Delete/Reactive Certificate Demonstration

Image 43 - Prototype Delete/Reactivate Certificate (part 1)

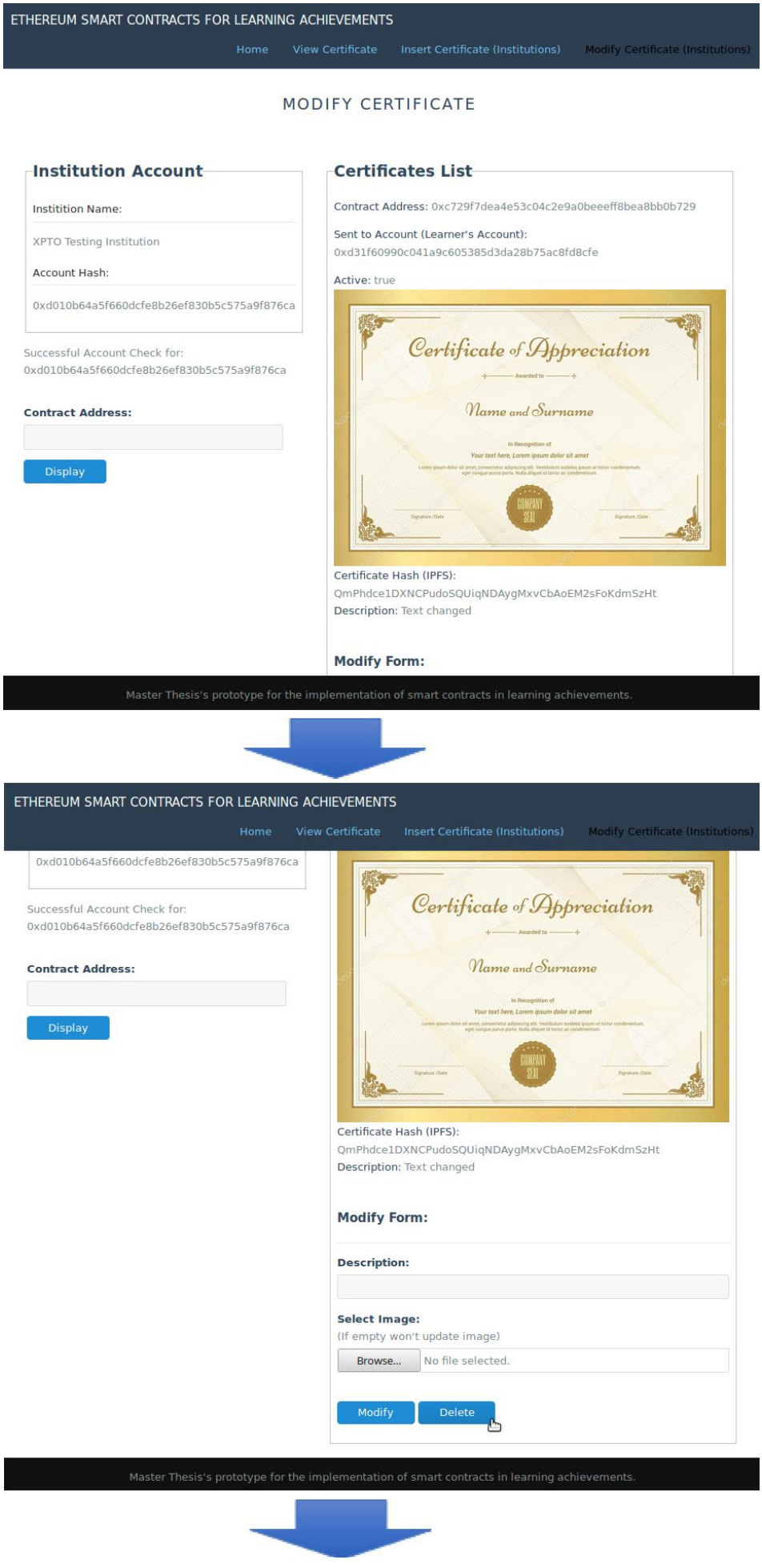


Image 44 - Prototype Delete/Reactivate Certificate (part 2)

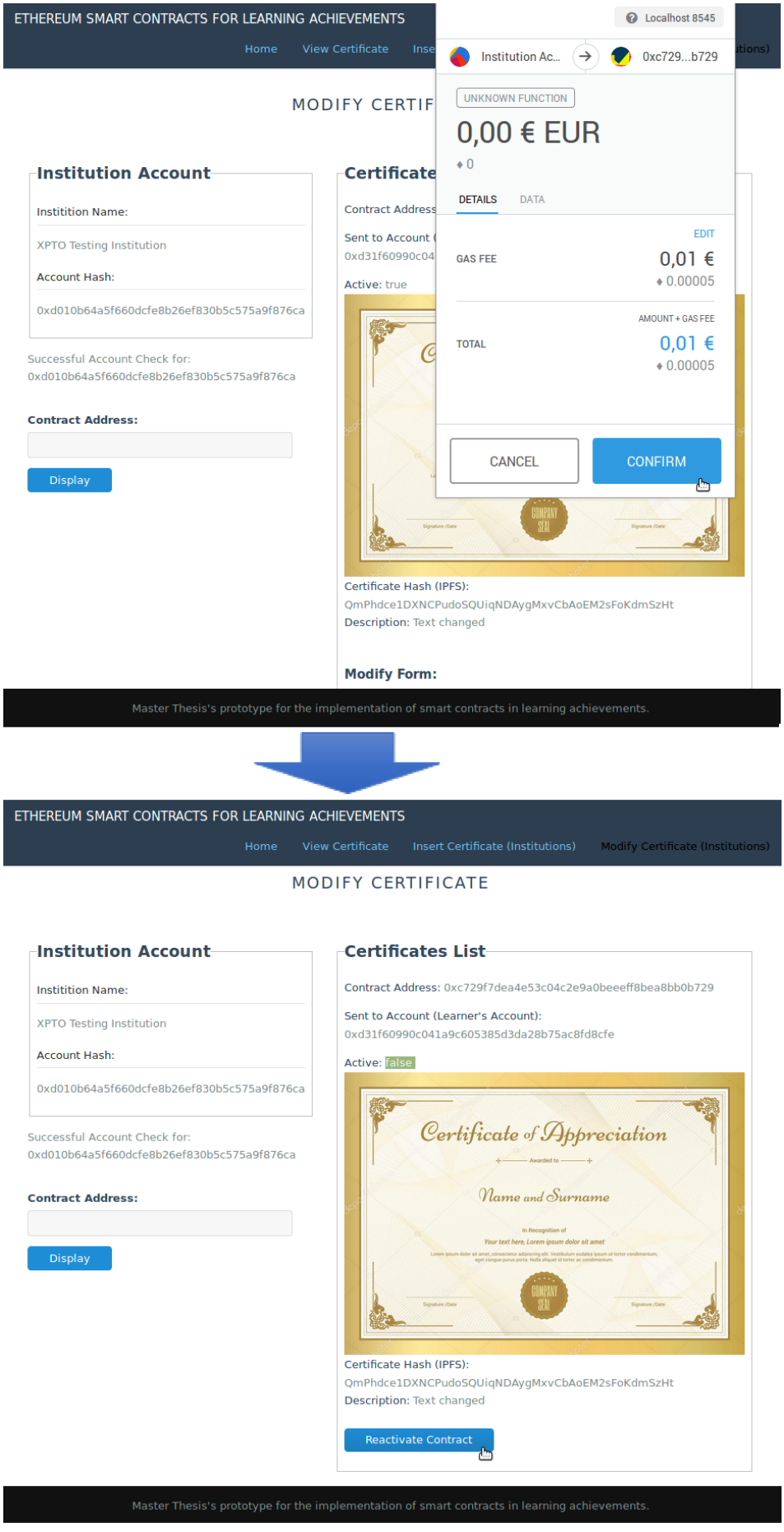


Image 45 - Prototype Delete/Reactivate Certificate (part 3)

Annex F.4 – View Certificate Demonstration (visibility management)

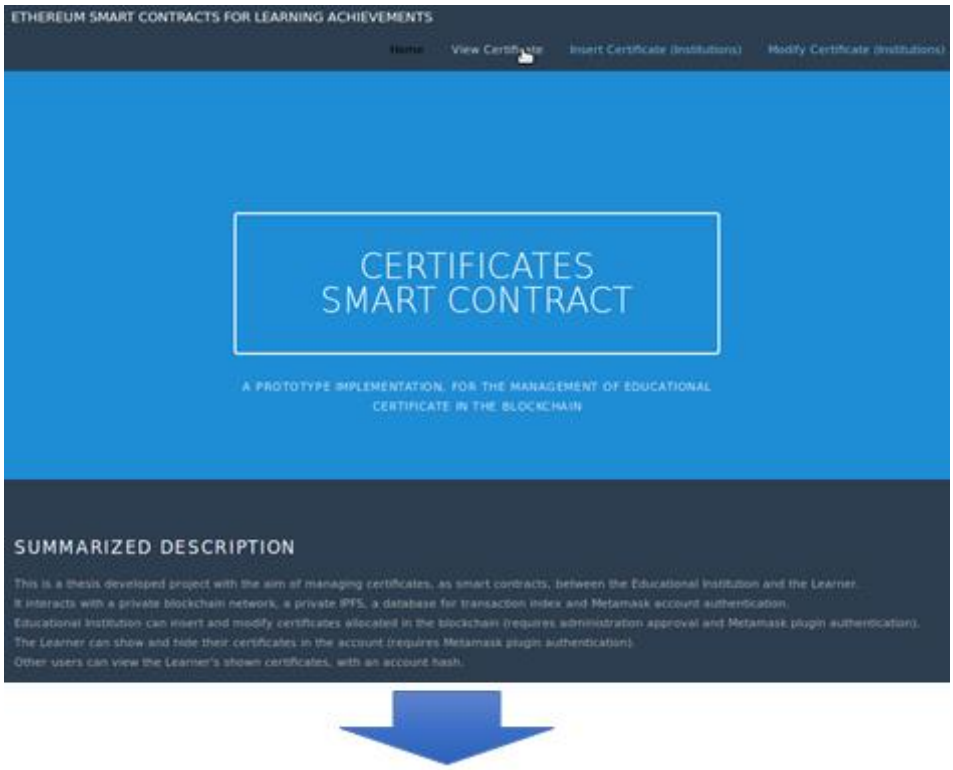
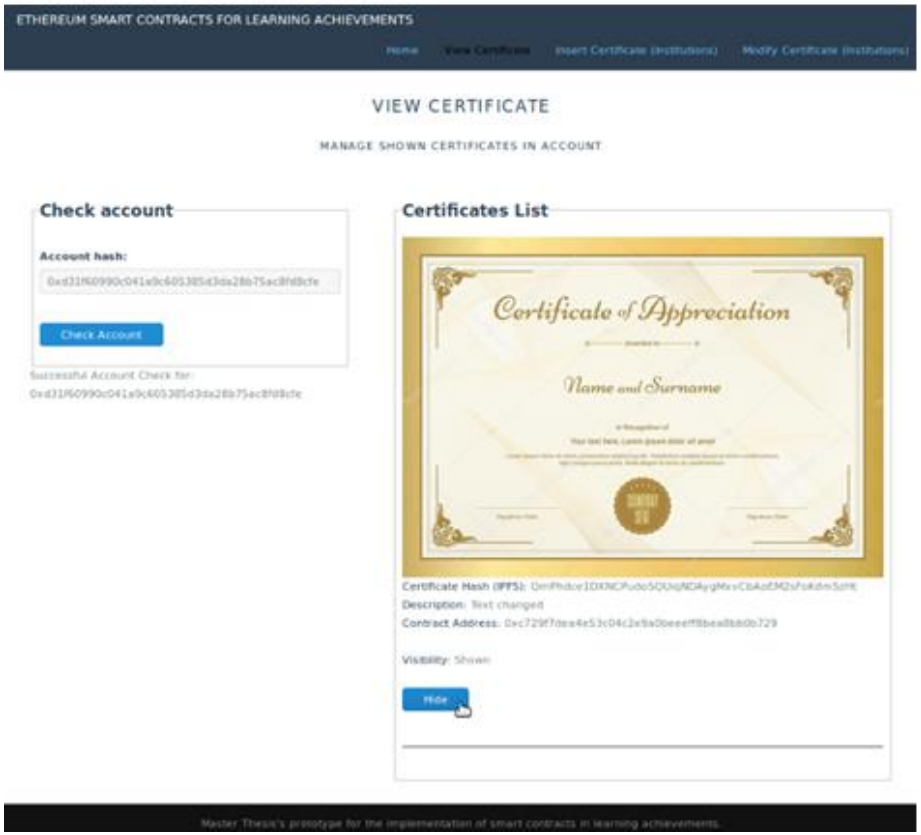


Image 46 - Prototype View Certificate (part 1)

Learner account



Other user (no account)

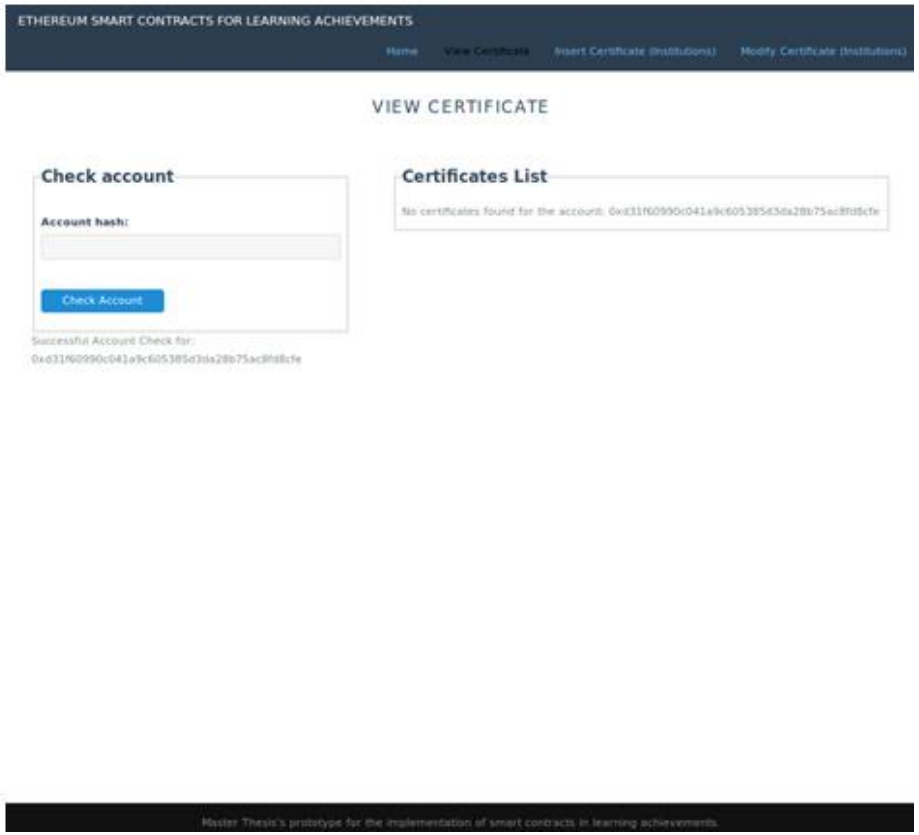


Image 47 - Prototype View Certificate (part 2)

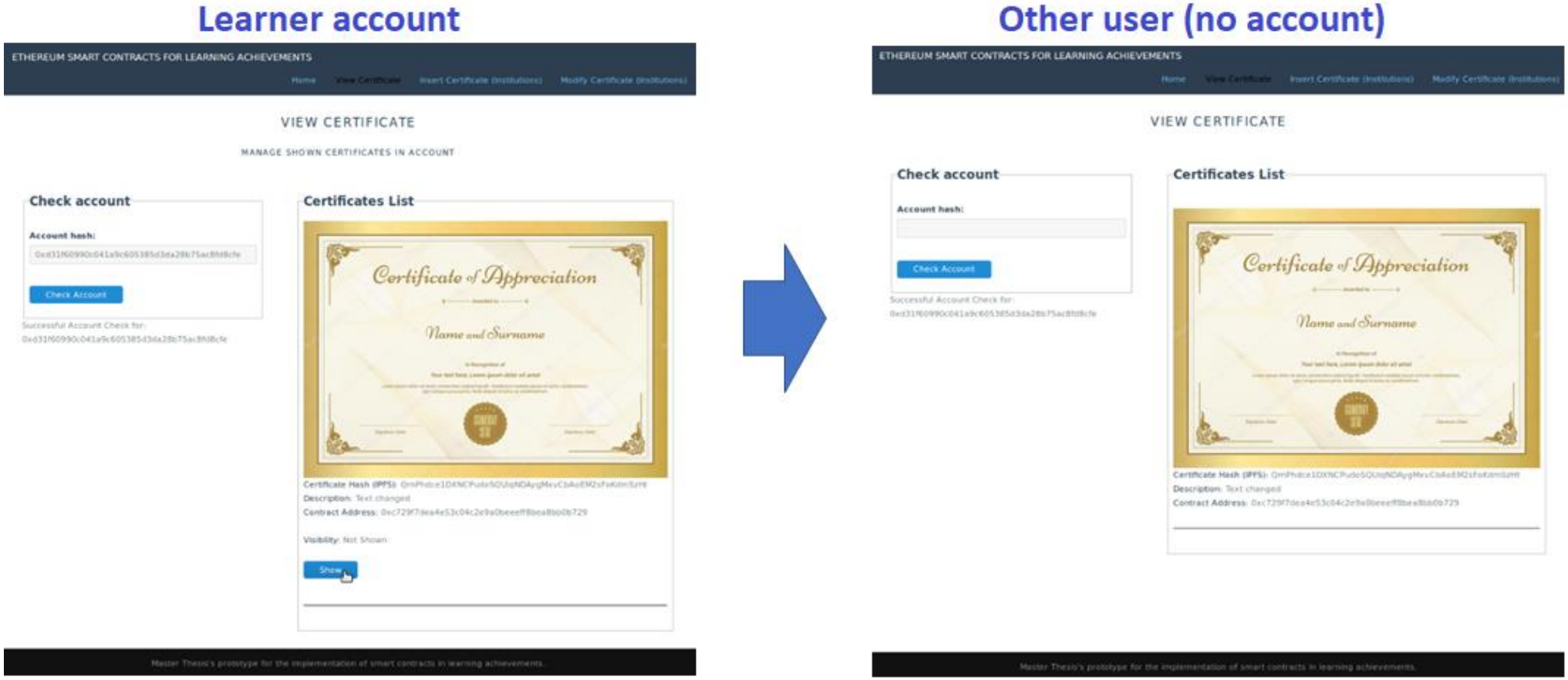


Image 48 - Prototype View Certificate (part 3)

Annex G - Secondary Tests and Experimentations

Conducted tests for the Certificates (alternative) and Login (optional) smart contracts. The tests are executed for the following functions:

- Certificate Alternative, composed by 2 certificates:
  - constructor()
  - newCertificate(_sendToAccount, _ipfsHash, _description)
  - setCertificate(_ID, _sendToAccount, _ipfsHash, _description)
- Login
  - constructor(accountHash, _accountName, _loginName, _loginPassword)
  - changeLogin(_loginName, _loginPassword, _accountName, _newLoginName, _newLoginPassword)
  - setActive(_loginName, _loginPassword, _active)

Table 11 show the tests parameters and Table 12 are the Experiments executed from the tests.

Table 11 - Secondary smart contracts Tests

Test	Method		Arguments	Gas Price	Gas Limit[47]
Certificate Alternative	**constructor**		none	1	937000
	1	**newCertificate**	_sentToAccount = 42 characters length _ipfsHash = 46 characters length _description = 100 characters length	1	298500
		setCertificate	_ID = "1" _sentToAccount = 42 characters length _ipfsHash = 46 characters length _description = empty	1	37500
		setActive	_ID = "1" _active = "false"	1	34000
	2	**newCertificate**	_sentToAccount = 42 characters length _ipfsHash = 46 characters length _description = empty	1	166000
		setCertificate	_ID = "2" _sentToAccount = 42 characters length _ipfsHash = 46 characters length _description = 100 characters length	1	157000
		setActive	_ID = "2" _active = "false"	1	34000
Login	**constructor**		_accountHash = 42 characters length _accountName = "Test" _loginName = "login" _loginPassword = "123"	1	976000
	changeLogin		_loginName = "login" _loginPassword = "123" _accountName = "Test New", _newLoginName = "login1", _newLoginPassword = "1234"	1	48900
	setActive		_loginName = "login1" _loginPassword = "1234" _active = "false"	1	36500

[47] "Gas Limit" values are the rounded value shown in Metamask confirmation transaction when issued a contract..

Table 12 - Secondary smart contract Experiments

Test	Method		Validation Time (Seconds)	Price Cost (Ether)	Price Cost (Euro)	Transaction Hash[48]
Certificate Alternative	constructor		26,72	0,00094	0,16	0xadbe5bfc814c71dedd8010b83a8ff411569c910b55174e7629736f8374b93262
	1	newCertificate	22,00	0,00030	0,05	0xc298055cd36e87164038c9adf674d4d0fab129f2549ca8c17dd0d7abc9370d9f
	1	setCertificate	11,61	0,00004	0,01	0xb184722bc53f30a3a00184317791b51faabe58901a76b6360acdec815a4a8e5d
	1	setActive	6,10	0,00003	0,01	0xb29054106cdafacf91c3071bc2c0fc17fa9daaff7ab2a8dde28702ce3aab8cc2
	2	newCertificate	13,49	0,00017	0,03	0x42d92f8b92db7cdc3f757fc57bd8550a7a3a6342fc3bd67690d7ff234428acc3
	2	setCertificate	24,98	0,00016	0,03	0xdecef83af17b1a0236e8b2d4114d75d1204caa4ec243681aa88fc8126ee74f9e
	2	setActive	27,55	0,00003	0,01	0xc7ad6997df30ce725f3a4c9fc4339eef5335b79b95462be70d7b039e5df46330
Login	constructor		22,27	0,00083	0,14	0x71cfb09c141210ff7d50a997916f02f239c17e72e1752f537a0fc02537b71950
	changeLogin		12,49	0,00005	0,01	0x4705e1fc62610fdbd2b1249897f98c3a9353e440663d1f69282c413b00d441f6
	setActive		21,45	0,00004	0,01	0xef0a7e1ed5fc3584eac9d06ce98b68efbd28509a12e9d79d03b299127c090b87

Experiments were executed on 12th October 2018, between 9:26pm and 11:17pm (+UTC).

These experiments don't show any relevant result discrepancy on the comparison between the used Certificate smart contract (Experiments section) to the Certificate Alternative. The only visible difference for the Certificate Alternative is that the certificates are stored in a single smart contract, saving at least 0,11€ per contract (comparing to Test E on the Table 9 - Experimentations Median Calculation) issued in an institution.

As for the Login smart contract, the project must prove if 0,14 Euros, per institution account hash, are proven useful for development and the costs are easily consented by the institutions, for the account's permission to use the project's application.

[48] Transaction hash on the Ropsten blockchain testing network. If information still persists on the testing network, more information can be checked using the web page Etherscan: https://ropsten.etherscan.io .