Universidade do Minho
Escola de Engenharia

Jorge Daniel Ribeiro Lopes

# Blockchain Technology Concepts and Applications

Dissertation of Integrated Master's degree in Engineering and Management of Information Systems

Dissertation developed under orientation of
Professor José Luís Mota Pereira

November 2018

# DECLRAÇÃO

Nome: Jorge Daniel Ribeiro Lopes

Endereço eletrónico: a73263@alunos.uminho.pt Telefone: 932051935

Número do Bilhete de Identidade: 14864954

Título da dissertação: Blockchain Technology Concepts and Applications

Orientador(es): Professor José Luís Mota Pereira

Ano de conclusão: 2018

Designação do Mestrado: Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para

prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respetiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 22/10/2018

Assinatura: Jorge Daniel Ribeiro Lopes

# ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor José Luís Mota Pereira, my dissertation supervisor, for giving me the opportunity to embrace this challenge, for his patient guidance and encouragement during this research work.

Without a doubt, most of the goals I have achieved in my life were not possible without my parents and my family, therefore I have to thank them for all their support.

I would also like to extend my thanks to all my friends for being with me in the good and bad moments of these years and for creating memories that I will treasure for eternity. I give special attention to Cristiana, for never leaving my side and always offer to help with every misfortune that has crossed my path.

I would also like to thank Praneeth Babu Marella from Boise State University, and Soumaya Ben Dhaou from UNU-EGOV, for taking their time to help me and share some of the knowledge they have regarding Blockchain.

# ABSTRACT

The increasing popularity of Blockchain technology has captured the attention of many industries and organizations. In simple terms, Blockchain is a distributed ledger technology that allows digital assets to be transacted in a peer-to-peer decentralized network, those transactions are verified and registered by every node of the network. Creating this way, a transparent and immutable history of records whose veracity is provided by the consensus protocol.

By enabling smart contracts to be deployed into a Blockchain platform, the number of possible use cases for this technology improves considerably. Eliminating the need for third parties and, therefore, allowing many processes, in both the public and the private sectors, to become more efficient and economical. In this document, some of these applications are described by presenting examples of projects already implemented or in the development stage

Although the rapid development of the technology, there are still a lot of limitations regarding its governance, scalability, and many other challenges, being them technical, legal or social-economic, that need to be overcome in order to achieve mass adoption.

To gain a comprehensive understanding of Blockchain technology and smart contracts, a proof of concept was developed, being the use case electronic voting systems. The objective is to develop a decentralized application as an example sufficiently demonstrative of the potential advantages of Blockchain solutions.

**Keywords:** Blockchain Technology, Smart Contracts, Blockchain Current Applications, Decentralized Application, Electronic Voting System.

# RESUMO

A crescente popularidade da tecnologia Blockchain tem captado a atenção de muitas indústrias e organizações. Em termos simples, Blockchain é um registo distribuído que permite que ativos digitais sejam transacionados de pessoa para pessoa em uma rede descentralizada, essas transações são verificadas e registradas por todos os nós da rede. Criando desta forma, um historico transparente e imutável de registros cuja veracidade é garantida pelo protocolo de consenso.

Ao permitir que smart contracts sejam implantados em uma plataforma Blockchain, o número de casos de uso possíveis para esta tecnologia aumenta consideravelmente. Eliminando a necessidade de entidades terceiras e, portanto, torna muitos processos, tanto no setor público quanto no privado, mais eficientes e econômicos. Neste documento são descritas algumas dessas aplicações, apresentando exemplos de projetos já implementados ou em fase de desenvolvimento.

Embora a tecnologia esteja a atingir um rápido desenvolvimento, ainda existem muitas limitações em relação à sua gestão, escalabilidade e muitos outros desafios, sejam eles técnicos, legais ou socio-econômicos, que precisam ser superados para alcançar uma adoção em massa.

Para obter uma compreensão abrangente da tecnologia Blockchain e smart contracts, foi desenvolvida uma prova de conceito, sendo os sistemas de voto electrónico o caso de uso. O objetivo é desenvolver uma aplicação descentralizada como um exemplo suficientemente demonstrativo para provar as vantagens potenciais das soluções Blockchain.

**Palavras chave:** Tecnologia Blockchain, Smart Contracts, Actuais aplicações da Tecnologia Blockchain, Aplicações Descentralizadas, Sistemas de Votação Electronica.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# LIST OF ACRONYMS

This document uses a set of acronyms, listed below:

AML – Anti Money Laundering

API – Application Programming Interface

BFT – Byzantine Fault Tolerant

B2C – Business to Consumer

Crypto – Cryptocurrencies

C2C – Consumer to Consumer

DAOs – Decentralised Autonomous Organisations

DLT – Distributed Ledger Technology

DPoS – Delegated Proof-of-Stake

DSR – Design Science Research

FBA – Federated Byzantine Agreement

HIE – Health Information Exchange

KYC – Know Your Costumers

MTO – money transfer operator

PBFT – Practical Byzantine Fault Tolerance

PoET – Proof of Elapsed Time

PoS – Proof-of-Stake

PoW – Proof-of-Work

# CHAPTER 1 - INTRODUCTION

This chapter introduces the subject of the study by presenting the research context and motivation for this dissertation, as well as the objectives, expected outcomes and the research methodology conducted. The last section of this chapter will be the structure of the document describing the contents of each chapter.

## 1.1. Research Context and Motivation

Blockchain can be referred to as the second era of internet, "The first era for decades was the internet of information. Now we're getting an internet of value. Where anything of value including money, our identities, cultural assets like music, even a vote can be stored, managed, transacted and moved around in a secure private way" (Don Tapscott, 2018).

The Blockchain technology was first conceptualized in October 2008 by an individual or a group under the name of Satoshi Nakamoto, as a core component for the Bitcoin cryptocurrency (Nakamoto, 2008). It was put into practice in 2009 by the same entity by releasing the version 0.1 of bitcoin software on Sourceforge, creating a transparent and secure network for digital payments, with immutable records and a distributed database.

This technology goes far beyond cryptocurrencies, it has indeed many applications that can influence significantly the way we interact with organizations and among ourselves. The purpose of this technology is the use of a distributed and decentralized ledger for verifying and recording transactions, allowing parties to send, receive, and record value or information through a peer-to-peer network of computers.

Notably, Blockchain has wide-ranging applications, including as a platform for smart contracts. When two parties agree on the terms and want to make a contract, they both digitally sign the contract. This contract will be programmed in a special code that contains the variables on which different outcomes are triggered. The contracts are self-enforced and will thus be executed exactly as coded beforehand (Tapscott & Tapscott, 2016). This enables a broad variety of uses cases and interactions that can be done using the Blockchain technology.

Although there are many advantages and opportunities provided by the Blockchain technology, there are also many issues and challenges that have been raised. It is important that the right balance is achieved between the rapid development of the technology and its governance, scalability and reliance on computational power in order to achieve mass adoption.

To better understand the Blockchain technology and smart contracts, it was developed a prototype, being the use case electronic voting systems, it is expected to be sufficiently demonstrative to prove the potential advantages of Blockchain solutions. It was developed on the Ethereum platform using the Solidity code language.

This dissertation was carried out in the scope of the Dissertation of the Integrated Master's degree in Engineering and Management of Information Systems, of the University of Minho, in Portugal.

## 1.2. Objectives and Expected Outcomes

This dissertation aims to identify and explain how the Blockchain technology work and its properties. It is also expected to explore the technology current applications and how can it be used to improve the way approach some use cases are approached. Additionally, a research must be conducted regarding the technology main limitations and challenges.

In order to consolidate the acquired knowledge, it is expected the development of a proof of concept in one of the available Blockchain platforms using a use case sufficiently demonstrative to prove the potential advantages of a Blockchain solution.

The use case must be analysed, searching for existing pain points and how can Blockchain improve or solve those problems. Regarding the development of the decentralized application, it is expected a description of the chosen Blockchain platform, smart contracts and any other components required for its good performance.

The purpose of this project is to gather and describe all the information required to create a starting point for everyone who wishes to explore this new technology.

## 1.3. Research Methodology

This section focuses on the research methodology followed during the execution of this dissertation. It comprises the research procedure and the research approach – Design Science Research (DSR), describing the phases that compound it. In the end, those phases were mapped according to this dissertation.

### 1.3.1. Research Procedure

In this section, it is presented the research procedure used during the literature review in order to achieve the principles of validity and reliability. Validity can be defined as the degree of precision in the identification and treatment of sources, including the selection of scientific databases, publications, the period covered, keywords, and the application of forwarding and backward search (Webster & Watson, 2002). Reliability refers to the replicability of the literature search and can be achieved through procedure documentation and the selection criteria used (Vom Brocke et al., 2009).

The literature search process consisted of the identification of relevant literature on the Blockchain domain. The literature search process was carried using keywords such as “Blockchain Technology”, “Blockchain Platforms”, “Hash Pointer”, “Mining”, “Consensus Protocol”, “Blockchain Industry Transformation”, “Blockchain ecosystem”, “Cryptocurrencies”, “Blockchain challenges and limitations” “Smart Contract”, “Solidity” and “Decentralized Applications”

The selection of the articles was based on the classification regarding the number of citations, and the reading of the abstracts to identify its relevance. In order to access, these papers scientific databases such as Google Scholar, Scopus and Science Direct were used. Another web platforms such as Medium, GitHub, Blockchain Revolution Weekly, and Blockchain Research Institute Newsletter were also very useful. In total, 81 papers were collected, from which 47 were considered useful for the development of this document.

During the realization of this dissertation two meetings regarding Blockchain and its applications were conducted, the first with Praneeth Babu Marella from Boise State University, and the second with Soumaya Ben Dhaou from UNU-EGOV. Those two interviews along with the Chain-IN 2018 conference, were also a great source of information.

### 1.3.2. Research Approach

Considering the purpose of this dissertation, it was decided to choose the Design Science Research (DSR). In this section, it will be described all the phases of this methodology and its reflection on this study.

DSR focuses on the utility of the artefact, an artefact can be a construct, model, method instantiation or a better theory (Kuechler & Vaishnavi, 2008). In this case, the artefacts are constructs, meaning the conceptual vocabulary of a domain, models, meaning a set of propositions or statements expressing relationships between constructs, and methods that mean a set of steps used to perform a task. As it can be seen in figure 1, DSR comprises six phases that must be followed during the design and evaluation of the proposed artefacts (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007).

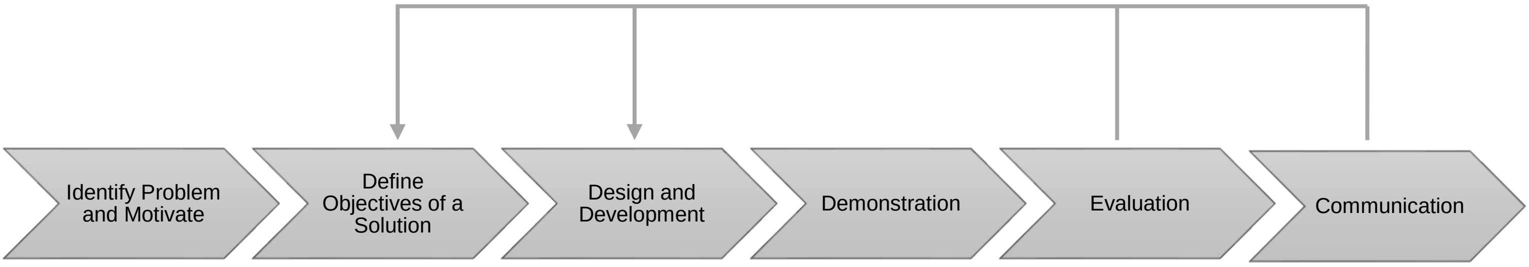

Identify Problem and Motivate
Define Objectives of a Solution
Design and Development
Demonstration
Evaluation
Communication

*Figure 1: DSR phases*

Identify Problem and Motivate: corresponds to the problem definition and justifies the value of a solution.

Since the definition of the problem will be used to the development of an artefact that can effectively constitute a solution, it is important to detail the problem in a way that the solution can capture its complexity. Justifying the value of the solution is not only to motivate the researcher that has been seeking for a solution, but also to help in the perception of the logic that leads the researcher to his understanding of the problem.

Define the Objectives of a Solution: based on the problem definition and knowledge of what is possible and feasible to attain, it seeks to infer the objectives of a solution.

The objectives can be either quantitative, quantifying certain terms that are the ideal solution, or qualitative, describing how can a new artefact represent a solution to the problems yet to be addressed.

For the successful accomplishment of this activity, it is expected as a requirement the knowledge of the state of the problem, the current solutions and their effectiveness, if they exist.

Design and Development: corresponds to the creation of the proposed artefact on the previous phase. This phase includes a transformation of the objectives into specific features of an artefact.

These artefacts can be constructs, models, methods or instantiations. The good performance of this activity is dependent on the theoretical knowledge that can be used to make the transition from the objectives to the design and development of the solution.

Demonstration: demonstrates the use of the artefact to solve one or more instances of the problem.

It can be done through experiments, simulations or other forms deemed appropriate. This activity requires prior effective knowledge of how to use the artefact to solve the problem.

Evaluation: corresponds to the observation and measure of how well the artefact supports a solution to the problem. This phase involves comparing the objectives of a solution to actual observed results from the artefact demonstration. At the end of this phase, the researchers can decide whether to iterate back to design phase or carry on to the next phase.

Communication: this phase concerns with communicating the problem and its importance, the artefact, its utility and innovation, the rigour of its design, and its effectiveness to researchers and other relevant audiences.

The success of this activity requires the knowledge on how to produce documents with scientific rigour and to identify opportunities that extend the reach of the work and value it to the scientific community. That said, this phase includes the writing and presentation of the dissertation, as well as the publication of related articles in relevant journals, magazines or conferences in the area.

## 1.4. Structure of the Document

This document is organized in six chapters, being the Introduction the first one where is described in the research context, motivation, objectives and expected outcomes and the research methodology used throughout this study.

The second chapter is Blockchain Technology, which focuses on the Blockchain and its different types, hash pointers and its tamper-evident properties, consensus protocols and how can they be compared between each other, smart contracts and the different programming languages that can be used to code them.

Furthermore, the third chapter is Current Applications of the Blockchain Technology, where examples of projects already implemented or in the development stage will be presented. It addresses a list of industries, indicating how can that specific sector be improved by adopting the technology.

The fourth chapter is Blockchain Limitations and Challenges, regarding the many different kinds of limitations that can prevent the technology from being globally adopted. Those limitations are technical, legal or socio-economic.

The fifth chapter is Blockchain Solution Example. It is composed of two sections, the first is a presentation of the chosen use case, regarding e-voting systems requirements and major problems to be solved. Being the second one the description of the developed decentralized application, specifying every component and explaining how it was developed.

Finally, the fourth and last chapter of this document is the conclusion, stating the main conclusions, systematizing the work that was carried out, the objectives achieved, the difficulties and limitations encountered, finalizing with the future work.

# CHAPTER 2 - BLOCKCHAIN TECHNOLOGY

In order to understand how the Blockchain technology work it is necessary to previously comprehend its key concepts. The purpose of this section is to synthesize all information gathered regarding Blockchain and its different types, hash pointers and its tamper-evident properties, consensus protocols and how can they be compared between each other, smart contracts and the different programming languages that can be used to code them.

## 2.1. Distributed Ledger Technology (DLT)

A ledger is, by definition, a book of records, keeping all the transactions of an organization or entity. Each transaction has a timestamp, allowing the ledger to be chronologically organized. This property makes the ledger an auditable history of all transactions made.

A distributed ledger is a synchronized and replicated database shared by all members of a network, its purpose is to record transactions made between members of the network (Brakeville & Perepa, 2016).

It is through a consensus protocol that the network participants govern and agree on which records should be updated into the ledger, this method eliminates the need for a centralized third-party mediator, such as a financial or government institution.

As it can be seen in figure 2, the Blockchain is one example of a distributed ledger technology, this means it also follows the properties mentioned above.

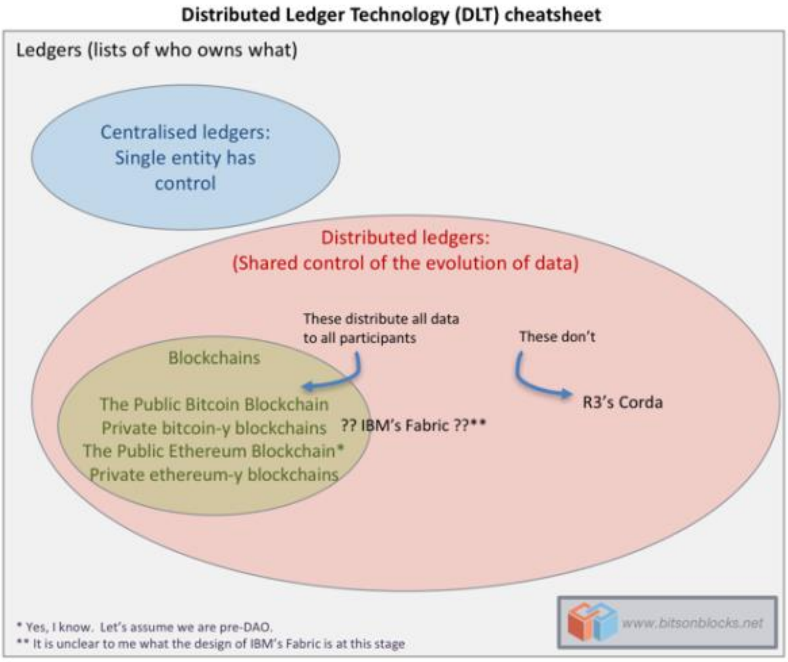

Distributed Ledger Technology (DLT) cheatsheet
Ledgers (lists of who owns what)
Centralised ledgers: Single entity has control
Distributed ledgers: (Shared control of the evolution of data)
These distribute all data to all participants
These don't
Blockchains
The Public Bitcoin Blockchain
Private bitcoin-y blockchains
The Public Ethereum Blockchain*
Private ethereum-y blockchains
?? IBM's Fabric ??**
R3's Corda
* Yes, I know. Let's assume we are pre-DAO.
** It is unclear to me what the design of IBM's Fabric is at this stage
www.bitsonblocks.net

*Figure 2: Digital Ledger Technology environment (Lewis, 2017)*

## 2.2. Blockchain Technology Concepts

Blockchain can be described as a distributed data structure that is replicated and shared among the members of a network, whose purpose is to record every transaction done in that specific network. Each transaction is batched into timestamped blocks and each block is identified by its cryptographic hash. Each block stores the hash of the previous one, creating a link between the blocks, or as the name implies, a chain of blocks (Christidis & Devetsikiotis, 2016). Blockchain was created in order to solve the double-spending problem, which means the result of successfully spending the same crypto or token more than once (Ofir Beigel, 2018).

To achieve a better understatement of Blockchain and how it works, it is recommended to separately describe each one of the major concepts of the technology.

Node: act as an entry point for one or more Blockchain users into the network.

Network: a group of nodes assembled together where each node contains an address.

Distributed network: a model in which components located on different devices from the same network are in constant communication and coordination in order to achieve a common goal. The key purposes of the distributed systems are resource sharing, openness, concurrency, scalability, fault-tolerance, and transparency.

Decentralized network: by relying on a peer-to-peer system, each party has the same capabilities and either party can initiate a communication session. Blockchain-based technologies allow each node to function as both a client and server. Disposing of the need for a central server or entity.

According to Vitalik Buterin, the co-founder of Ethereum, "Decentralization is one of the words that is used in the crypto economics space the most frequently and is often even viewed as a Blockchain's entire "raison d'être", but it is also one of the words that is perhaps defined the most poorly. Thousands of hours of research, and billions of dollars of hashpower, have been spent for the sole purpose of attempting to achieve decentralization, and to protect and improve it, and when discussions get rivalrous it is extremely common for proponents of one protocol (or protocol extension) to claim that the opposing proposals are "centralized" as the ultimate knockdown argument" (Vitalik Buterin, 2017).

Timestamp: An exact time when a block was generated, used to implement a time order chain structure. This time unit follows the Unix time system.

Chain: A group of blocks linked together by hash pointers.

Blocks: Blocks are composed by a list of transactions recorded over a given period and a header. The header of a block contains different sets of metadata, identifying the block version number, the previous block hash, the block timestamp, the nonce, the difficulty of mining the block and the last set is a Merkle tree root (Jiang et al., 2018).

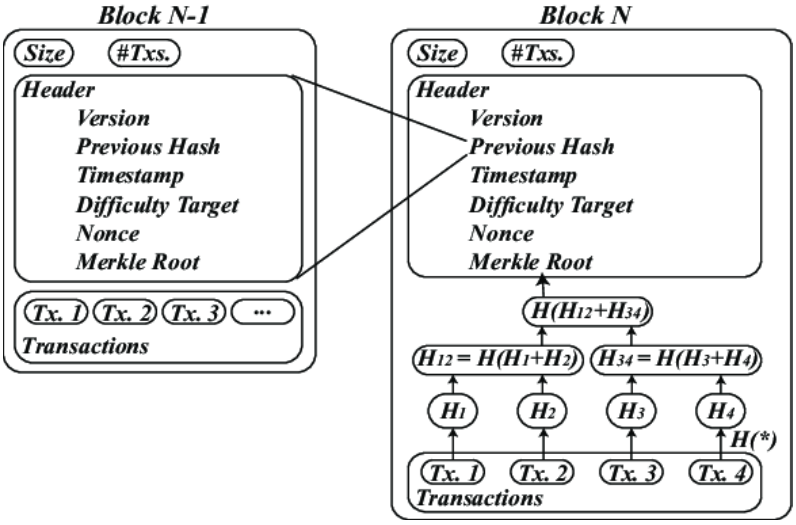


*Figure 3: Structure of a block (Jiang et al., 2018).*

As mentioned above, the header of a block (figure 3) is composed of the previous Hash, that is the result of encrypting all data found on the previous block header. The nonce, that is a value adjusted by miners so that the hash of the block generated will be bellow or equal to the current target of the network. The difficulty, that is a number that regulates how long it takes for miners to add new blocks of transactions to the Blockchain. The Merkle tree root summarizes all the transactions in the block, making possible to verify if a transaction is included or not.

Transactions: Transfer of value in specific cryptocurrencies or token. After a suitable confirmation, the transaction is included in a block and validated. To make a transaction the user needs the address (public key) of the receiver user wallet and his own private key, a digital signature only known by the owner of that wallet (Christidis & Devetsikiotis, 2016). In figure 4, it is presented an example of a transaction.

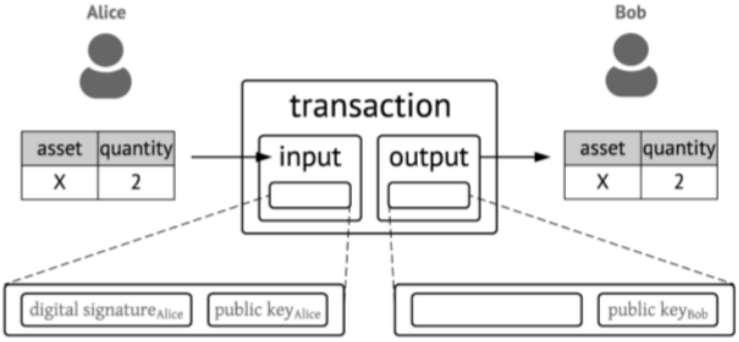


*Figure 4: Example of a transaction (Christidis & Devetsikiotis, 2016)*

To ensure that a transaction is valid and should be preserved on the Blockchain some confirmations are executed, such as:

- Confirming that the proposed address exists;
- Confirming the validity of the provided signature;
- Confirming if the entity who created the transaction effectively owns the aster about to be transacted;
- Confirming that the asset about to be transacted was not previously spent.

Each transaction as a small cost called "fee", the purpose of this fee is to reward the miner of the block in question. The larger the fee, more it will be the incentive to the miner to select this transaction.

## 2.3. Types of Blockchain

According to Vitalik Buterin, a Blockchain can be categorized into three different types, being them public, private and consortium Blockchains. They differ on user's restrictions to interact with the network (Buterin Vitalik, 2015).

Public Blockchains: a public or permissionless Blockchain has no restrictions regarding who can interact with it. It is open for everyone and every user can send transaction and participate on the consensus process. These Blockchains are generally considered to be fully decentralized.

Private Blockchains: in a private or permissioned Blockchain, write permission are centralized to one organization, read permission can be public or restricted, depending on the organization goals and if they desire a public auditable Blockchain.

Consortium Blockchains: a consortium Blockchain combines private and public network properties, the consensus process is usually controlled by a pre-selected set of nodes, being this party responsible for validating all blocks of the network. The read permissions, like private Blockchains, can be public or restricted. These Blockchains are generally considered to be partially decentralized.

### Comparison of different Blockchain types

In an attempt to compare and classify different types of Blockchains, it will be used a set of tables that, according to Voshmgir and Kalinov (2017), can help to distinguish between each type.

In table 1, differs Public from Private Blockchains, being the evaluation criteria Access, Speed, Security, Identity and Asset. Regarding security, the mentioned consensus mechanisms are described in the following pages.

| | **Public** | **Private** |
|---|---|---|
| ***Access*** | Open read/write | Permissioned read and/or write |
| ***Speed*** | Slower | Faster |
| ***Security*** | Proof of Work<br>Proof of Stake<br>Other consensus Mechanisms | Pre-approved participants |
| ***Identity*** | Anonymous<br>Pseudonymous | Know identities |
| ***Asset*** | Native Asset | Any Asset |

*Table 1: Public vs Private Blockchain adapted from (Voshmgir & Kalinov, 2017)*

In table 2, differs Public from Private and Consortium Blockchain, being the evaluation criteria the Participants, Consensus Mechanisms and Transaction Approval Frequencies.

| | Public | Consortium | Private |
|---|---|---|---|
| *Participants* | Permissionless<br>- Anonymous<br>- Could be malicious | Permissioned<br>- Identified<br>- Trusted | Permissioned<br>- Identified<br>- Trusted |
| *Consensus Mechanisms* | Proof of Work,<br>Proof of Stake<br>- Large energy consumption<br>- No finality<br>-51% attack | Voting or multiparty consensus algorithm<br>- Lighter<br>- Faster<br>- Low energy consumption<br>- Enable finality | Voting or multi-party consensus algorithm<br>- Lighter<br>- Faster<br>- Low energy consumption<br>- Enable finality |
| *Transaction Approval Freq.* | Long<br>-Bitcoin: 10 min or more | Short<br>-100x msec | Short<br>-100x msec |

*Table 2: Public vs Consortium vs Private Blockchain adapted from (Voshmgir & Kalinov, 2017)*

## 2.4. Hash Pointer

By cryptography is meant the study and practice of securing private messages so they can only be read by entities with permission to do so. It involves the encrypt and the decrypt of content using various mathematical equations and encryption keys. Modern cryptography has grown to include many other fields of study such as data integrity and user authentication.

### Hash Function

There are several different classes of cryptographic hash functions, although they are all forced to satisfy certain properties to achieve their best potential ("Blockgeeks," n.d.)

### Property 1: Computationally Efficient

For a cryptographic hash function being computationally efficient means that computers must be able to perform a hash function’s mathematical labour in a short period of time.

This property addresses the fact that if the hash function requires a large amount of time to process and deliver the solution, then it would no longer be useful or practical.

### Property 2: Deterministic

Cryptographic hash functions should always present the exact same result to the same given input. If a hash functions produce different outputs for the same input, then the results would be a random value, making it impossible to verify if a specific input is valid. Therefore, if a cryptographic hash function is not deterministic it will be useless.

### Property 3: Pre-Image Resistant

For a cryptographic hash function being pre-image resistant means that the output produced by the hash must not reveal any information about the input. Regardless of the size or data type of the input, the output must be always a fixed-length alphanumeric code. Any slight difference in the input is going to entirely change the output

This property reassures the privacy and security of the input given, making impossible to predict if was long or short, numbers or letter, etc.

### Property 4: Collision Resistant

For a cryptographic hash function being collision resistant means that it must be extremely unlike or even practically impossible, to two different inputs produce the same output.

The input of a hash function can have any length, although outputs have a fixed-length, this means that there are a limited number of solutions that can be produced. Since the number of inputs is infinite but the outputs are restricted, then it is a mathematical certainty that more than one input produces the same output.

So, in order to be collision resistant, the hash function has to guaranty that the probability of producing the same output from different inputs is so astronomically improbable that the possibility can be practically dismissed.

### Property 5: Puzzle Friendly

For a cryptographic hash function being puzzle friendly, let's call it "H", means that for every output "Y", if "k" is chosen from a high min-entropy, then it is infeasible to find an input "x" such that H(k|x) = Y.

High min-entropy means that the distribution is so spread out that no particular value is chosen with more than negligible probability.

Examples of cryptographic hash functions:

- Secure Hashing Algorithm (SHA-2 and SHA-3), the algorithm used by bitcoin, SHA-256, belong to this class.
- Keccak-256, algorithm currently used by Ethereum.
- Message Digest Algorithm 5 (MD5)
- BLAKE2

## Hash Pointer

A data structure is a specialized way of storing data, it can be stored as pointers or Linked List. Pointers are variables that store an address to another variable, while linked lists are a sequence of blocks that store data and are linked by pointers ("E-learning Spot," n.d.)

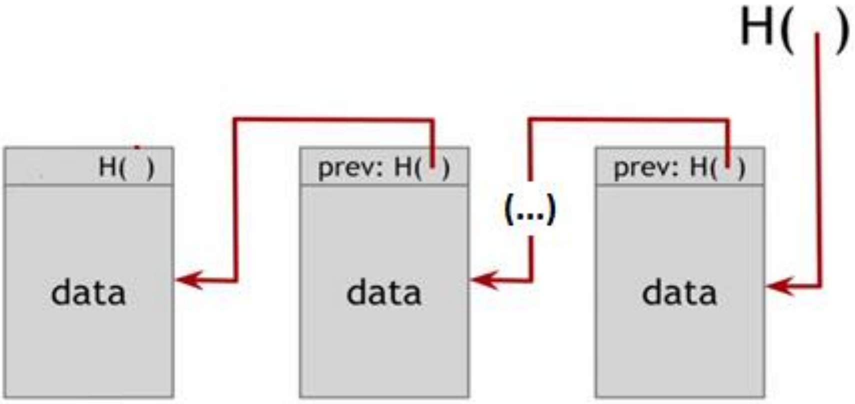


*Figure 5: Example of a Blockchain ("E-learning Spot," n.d.).*

A Blockchain is a linked list, linked by hash pointers (figure 6), where each block contains a hash pointer that stores a cryptographic hash of the header of the previous block in the chain.

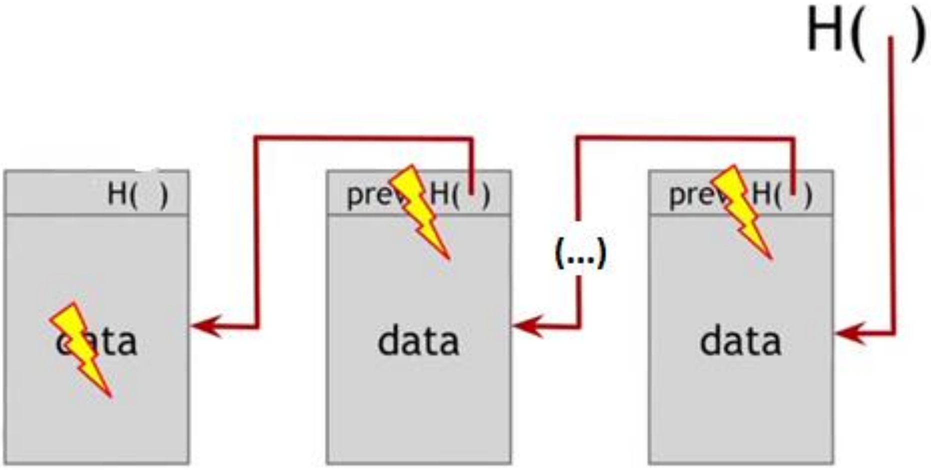


*Figure 6: Example of a corrupted Blockchain ("E-learning Spot," n.d.).*

The hash pointer provides the Blockchain with tamper-evident properties in a simple way. For example, if someone changes the contents of one block (figure 7) the hash of the next block will not mash up and inconsistency will be perceived. Even if the attacker has enough computer power to corrupt one block and all the following blocks in the chain, he will arrive a dead-end because the last hash pointer is the value remembered by all nodes on the network as being the head of the list and the inconsistency will be noticed inevitably.

## 2.5. Consensus Protocol

The consensus protocol is the mechanism that allows a decentralized network to arrive at an agreement about the state of the Blockchain and forces all nodes to behave accordingly to the network principles. These are the proprieties that eliminates the need for a regulator entity on a Blockchain network (Ouattara, Ahmat, Ouédraogo, Bissyandé, & Sié, 2018).

Next, it will be explained the most common models, explaining how they achieve consensus and highlighting some of their strengths and weaknesses.

Proof-of-Work (PoW): consists in solving a difficult cryptographic puzzle and broadcast the results to the network, that add the newly generated block to the chain. The participants who compete to solve this puzzle are usually called "miners", by solving this the miner usually receive as a reward an amount of crypto that is created along with the block and a small transaction fee.

PoW it was the first consensus protocol to be used and is still the most popular, but it has some major disadvantages like the need for a large amount of resources such as computing power and energy, is also vulnerable to 51% attacks, meaning that if a miner or a poll gather more than 50% of the network computing power the Blockchain can be manipulated in their behalf.

Proof-of-Stake (PoS): forces the participants to compete not by computing power but with cryptocurrencies instead, PoS algorithm randomly selects a validator for the block creation. The participants can increase their chances to be the one validating the block by increasing the amount of crypto they put on stake. Validators are paid only in processing fees.

PoS is a solution to the PoW energy consumption and computing power problem, but this also means that to corrupt the network the participant just needs enough money to invest. Like the PoW centralization risk, this is also a possibility in PoS, where the richest stakeholders can have control of the consensus in the Blockchain.

Delegated Proof-of-Stake (DPoS): works in a similar way to PoS, the participants can use their balances to elect delegates, called witnesses, and only this witness has the opportunity to put cryptocurrencies on stake to be the one validating the block. If a malicious action is made by a witness it can be detected by the one who elects him. When a block is validated, all wallets who elect that participant as a witness will receive a reward along with the validator.

These adjustments to PoS theoretically increase the distribution of rewards and real-time voting security, but like in PoS it still vulnerable centralization, there are fewer participants in charge of validating blocks, making it easier for witnesses to organize between them and corrupt the network.

Byzantine Fault Tolerant (BFT): every transaction needs to be signed to be verified each time it passes through a node. If a majority of responses are identical, then the network agree that the transaction is valid.

If the number of nodes of the network is very large it will increase the required number of transactions sent between them resulting in an overhead, this means that has low scalability properties.

Practical Byzantine Fault Tolerance (PBFT): follows the concept of the replicated state machine and voting by nodes to state changes. It also provides several important optimizations, such as signing and encryption of messages exchanged between nodes and clients, reducing the size and number of messages exchanged, for the system to be practical in the face of Byzantine fault.

This approach imposes a low overhead on the performance of the replicated service, however, like BFT, has only been scaled and studied to 20 nodes. Its messaging overhead increases significantly as the number of replicas increase.

Federated Byzantine Agreement (FBA): relies on a small set of trusted parties to achieve consensus, assuming that a given node knows a subset of nodes in the network and that the network agrees to accept the information provided by that party as true. The consensus is then formed as the members of the party form a collective agreement on the information.

Proof of Elapsed Time (PoET): every node has to request a wait time from a trusted function, this function is stored in a processor chip. The node with the shortest wait time for a given transaction block is elected the leader. Like other consensus algorithm presented before, the leader role is chosen randomly but the probability to be selected is proportional to the resources contributed.

This protocol decreases the costs in computing power and energy but the trust in the network is directly related with the trust on the chip maker and the function it provides.

Proof-of-Activity (PoA): combines PoW and PoS, similar to PoW, cryptographic puzzles need to be solved by miners, when the new block is mined, the system switches to PoS, where a participant is randomly chosen in a group to validate or sign the block. as in PoS, the participant with more cryptocurrencies on stake has more chances to be the one validating the block.

The issues in this protocol are the same found in PoW and PoS.

Proof-of-Burn (PoB): the participants have to send their cryptocurrencies to an address from where they cannot be recovered, by doing that they grant the right to miner blocks in proportion to the coins burnt.

This protocol, although it doesn’t consume as much energy as PoW, it still wastes resources and is also vulnerable to centralization.

Proof-of-Capacity (PoC): provide nodes with the ability to use empty space in their hard drive to increase the chance to mine available blocks, by storing a list of possible solutions on the mining device’s hard drive even before the mining activity begins. The larger the hard drive, the more possible solutions can be stored, the more chances a miner has to match the required hash value from his list, resulting in more chances to win the mining reward.

This protocol has a lower adoption rate and is vulnerable to malware affecting mining activities.

### Comparing consensus protocols

According to the description of the consensus protocols, they will be measured (table 3) using the same parameters that were chosen to better distinguish their benefits and disadvantages (Baliga, 2017). PoA, PoB and PoC are the least used protocols among the others so are excluded from the comparison.

| | PoW | PoS | PoET | BFT variants | FBFT |
|---|---|---|---|---|---|
| *Blockchain type* | Permissionless | Both | Both | Permissioned | Permissionless |
| *Transaction finality* | Probabilistic | Probabilistic | Probabilistic | Immediate | Immediate |
| *Transaction rate* | Low | High | Medium | High | High |
| *Token require* | Yes | Yes | No | No | No |
| *Cost of Participation* | Yes | Yes | No | No | No |
| *Scalability* | High | High | High | Low | High |
| *Trust model* | Untrusted | Untrusted | Untrusted | Semi-trusted | Semi-trusted |
| *Adversary Tolerance* | <=25% | Depends on specific algorithm used | Unknown | <=33% | <=33% |

*Table 3: Comparing different consensus protocols adapted from (Baliga, 2017).*

Blockchain Type: indicates the type of Blockchain platform where the consensus model is implemented. It can be Permissioned, Permissionless or able to be implemented in both platforms. As described on table 3, PoW and FBFT models are usually implemented in public and permissionless platforms, technically they can be used in permissioned platforms but would get a poor performance. As for BFT and variants is recommended a permissioned platform.

Transaction finality: indicates whether the transaction added to a block is considered final or not. In a probabilistic transaction finality, in models like PoW and PoET, it exists the risk to multiple blocks being mined at the same time due to their process of electing a leader and the network latencies, generating this way temporary forks that will be rejected posteriorly. This means that the time for a transaction to be confirmed and finalized is significantly increased. In models with immediate finality like FBFT, as soon as the transaction is included in the block, it is confirmed.

Transaction rate: indicates how fast can a transaction be validated. Probabilistic models are usually requiring more time, like PoW where miners have to spend a significant amount of time-solving cryptographic puzzles. PoET has a higher transaction rate comparative to PoW because of his leader selection mechanism. The other consensus models can confirm transactions in a much smaller amount of time, so it supports high transaction rates.

Token require: indicates if the model requires a token to achieve consensus or as an incentive. Models like PoW and PoS are based on the existence of a cryptocurrency, the other three do not require tokens to achieve consensus.

Cost of participation: indicates if it is required some kind of investment to participate on the network consensus. In PoW it is necessary to expend energy to mine blocks, while in PoS it is required to buy some initial cryptocurrencies to be able to stake in order to validate a block.

Scalability of peer network: indicates the ability to reach consensus when the number of peer nodes is constantly increasing. All models except BFT and variants have high scalability, for BFT the recommended number of nodes is 20.

Trust model: indicates if the nodes of the network have to be known or trusted. In models like PoW, PoS and PoET it is not necessary to know the identity of the node because the consensus is achieved by other means like computational work or security deposits. In BFT and FBFT, it requires knowing the identity of peering nodes to execute consensus decisions.

Adversary tolerance: indicates the percentage of the network that may act maliciously to the network without the consensus being compromised.

## 2.6. Smart Contracts

A smart contract can be described as an autonomous agent stored in the Blockchain. Once deployed, the contract will be identified by an address, it also contains the associated code, a balance, and its own private storage. Its private storage will allow the code to manipulate variables like in traditional imperative programs (Luu, Chu, Olickel, Saxena, & Hobor, 2016).

By sending a transaction to the contract address the contract will be triggered and will execute its code independently and automatically in a prescribed manner on every node in the network, its response will depend on the data included in the triggering transaction (Christidis & Devetsikiotis, 2016). The smart contract will follow an "If This Than That" logic, meaning that, it will react to actions made by participants or another smart contract. Each smart contract should be coded in a deterministic way, meaning that the same input should always produce the same output and preferably all possible outcomes should be previously predicted.

Smart contracts allow us to have general purpose computation occur on the chain, although, their main purpose is to manage data-driven interactions between entities on the network. It is not required for a smart contract to be run on a Blockchain, but by doing so inherits its trust properties, building confidence to interact whit unknow and untrusted people and still be sure they will be forced to follow the same contractual clauses and that these clauses cannot be changed after the smart contract has been deployed on the Blockchain.

In reality, the concept of smart contract was studied long before the Blockchain technology, by Nick Szabo in 1997. The real-life example he used was a vending machine, where customers can interact with it by putting coins and choosing a product, the code implemented on the machine will react by dispensing the change and the product according to the established price (Szabo, 1997).

### Smart Contract Coding

The most widely used smart contract programming language is Solidity, but it is not the only one. Next, it will be compared some of these languages and highlighting some key features (Parizi, Amritraj, & Dehghantanha, 2018).

Solidity: executed on the Ethereum Virtual Machine (EVM), it has a similar syntax to JavaScript, it is a statically typed language that supports inheritance, libraries and complex user-defined types.

Pact: it is a turing-incomplete safety-oriented programming language, that allows the execution of mission-critical business operations quickly and safely way, making bugs harder to write and easier to spot.

Liquidity: it is a high-level, fully-typed functional language providing three main features, such as coverage of the Michelson language that was the programming language for smart contracts executed on Tezos before the creation of Liquidity, local variables instead of stack manipulations and can define high-level types.

Simplicity: higher-level programming language that allows you to write more human-readable smart contracts. By abstracting many of these low-level concepts from Bitcoin Script, Simplicity makes it faster and easier than ever to write smart contracts.

Rholang: a functional language that offers many of the same capabilities as Simplicity and Solidity but operates in a specifically functional context, created for blockchain developers who prefer to work in a functional programming environment.

Although Solidity is the most used language due to his high usability properties, it still has some vulnerabilities. We should have in mind that all these programming languages are still improving and susceptible to changes.

### Security bugs in Contracts

There are several security bugs, that when found by malicious miners or users, can be exploited to gain profit (Luu et al., 2016). Some of those bugs will be described next.

### Transaction-Ordering Dependence

Considering a scenario where the Blockchain is at state $\alpha$ and a new block contains two transactions invoking the same contract, the users will have uncertain knowledge of which state the contract is at when their individual invocation is executed.

This situation can be problematic for two reasons, the contract can yield an unexpected result if there are concurrent invocations, secondly, a malicious miner or user can exploit this situation to gain profit or even steal money.

### Timestamp Dependence

In some cases, the contracts condition to execute come critical operations are triggered by the block timestamp. Normally, the timestamp is set at the current time of the miner's local system, however, the miner has the flexibility to vary this value by roughly 15 minutes (regarding Ethereum platform). A malicious miner can use this flexibility to choose a specific timestamp to manipulate the outcome of this kind of contracts.

### Reentrancy Vulnerability

If a contract sends a call to another one, the current execution waits for that call to finish, this situation can become problematic when the recipient of the call makes use of the intermediate state the caller is in. If the code does not take this situation into account, a malicious user could exploit it by continuing addressing the contract while it is still waiting for the call to finish.

# CHAPTER 3 - CURRENT APPLICATIONS OF THE BLOCKCHAIN TECHNOLOGY

This chapter will focus on the current applications of the Blockchain technology, by presenting examples of projects already implemented or in the development stage. It addresses a list of industries, indicating how can that specific sector be improved by adopting the technology.

Don Tapscott, Co-founder and Executive Director at Blockchain Research Institute, describes Blockchain as follows.

"At its most basic, Blockchain is a global database, an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions, but virtually everything of value and importance to humankind: birth and death certificates, marriage licences, deeds and titles of ownership, educational degrees, financial accounts, medical procedures, insurance claims, votes, transactions between smart objects, and anything else that can be expressed in code. This ledger represents the truth because mass collaboration constantly reconciles it"(Tapscott & Tapscott, 2016).

According to his words, Blockchain is a versatile technology and can be implemented in a vast list of industries (figure 8) and maybe change completely the way some of their use cases are approached.

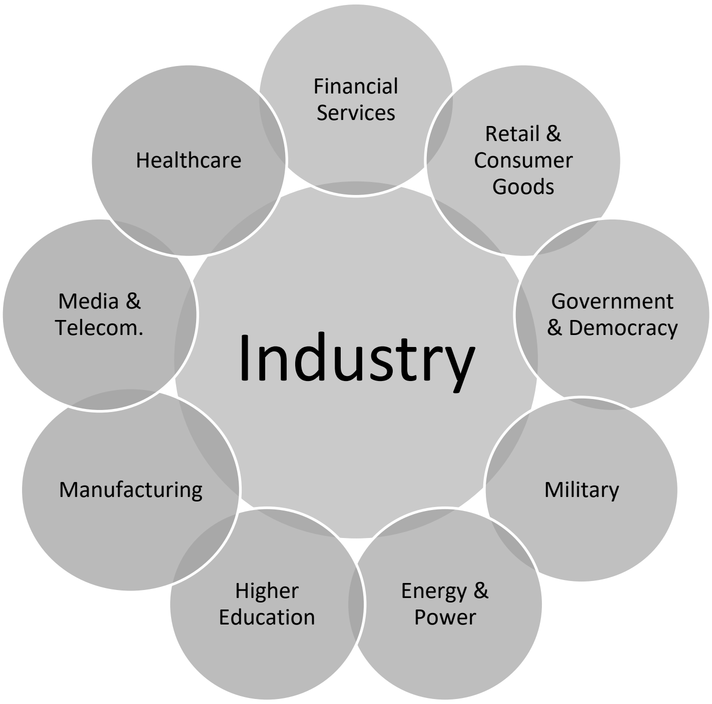


*Figure 7: Industries to transform*

## 3.1. Financial Services

The finance business sector is the main focus of Blockchain projects. It can be divided into six categories, figure 9, being them Payments, Insurance, Deposits & Lending, Capital Raising, Investment Management, and Market Provisioning.

On each one of these categories, some use cases are addressed that can be improved using Blockchain. Additionally, the potential effects which might be achieved when certain conditions are met, are also presented.

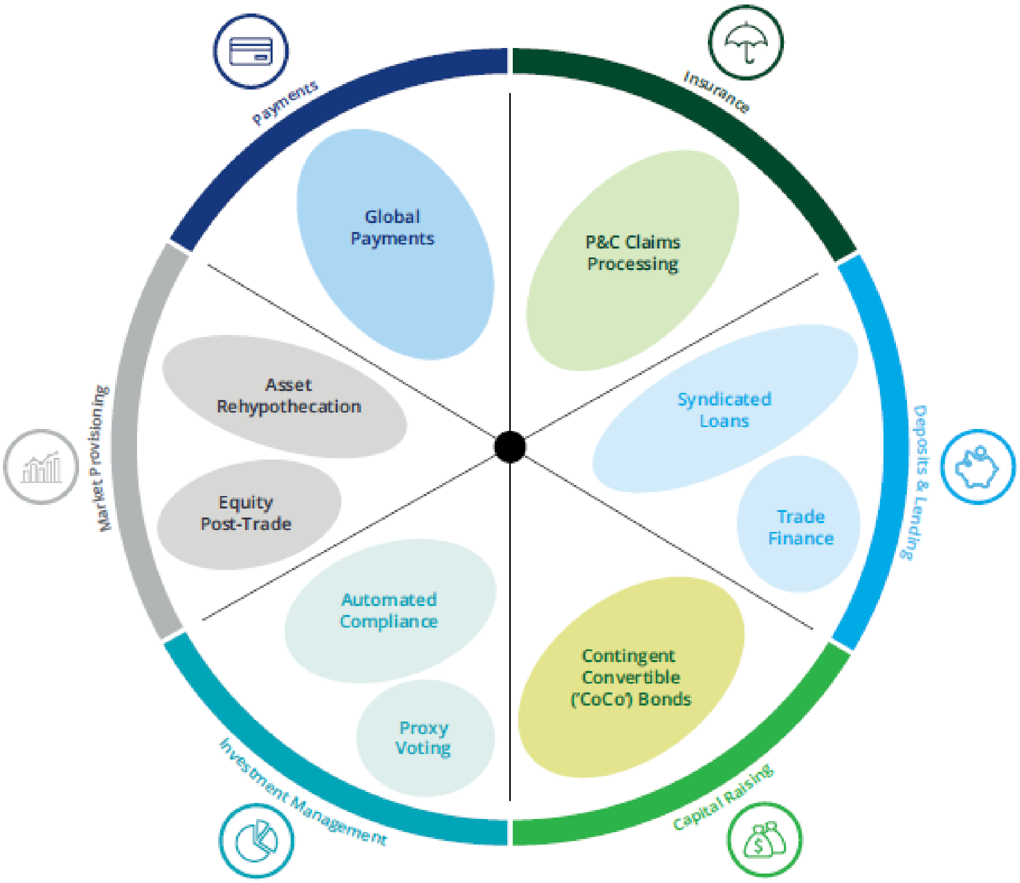


*Figure 8: Blockchain use cases in Financial Services (Contri & Galaski, 2016)*

### Commercial property & casualty claims processing

Claim and loss processing is one of the main sources of friction in the Commercial property and casualty insurance market. The reason for this lies in the standard process to submit a claim. Usually, the insured must complete a complex questionnaire accompanied by the receipts of all costs, this process is mediated by brokers which can create delays.

On the other hand, insurers must rely on third parties to provide asset, risk and loss data in the adjudication and underwriting process. Collecting the data is a manual effort and in some case is impossible to have access to the information or it may not be updated. By using Blockchain it is possible to reduce fraud and improve assessment through historical claims information (Contri & Galaski, 2016).

#### Benefits

- Customer-friendly: Smart contracts and smart assets will remove manual effort from the claim submission process;
- Direct: Blockchain will share loss information among insurers, eliminating the need for brokers;
- Practical: Loss adjusters will no longer have to review every claim, except in specific risk situations;
- Clean: Insurers will have seamless access to historical claims and asset provenance, making it easier to spot suspicious behaviour;
- Integrated: The Blockchain will automatically combine data sources from trusted providers;
- Fast: In most cases, smart contracts will facilitate payment without involvement from the back office.

| Potential effects | Necessary conditions |
|---|---|
| **Claims** are processed automatically using trusted data sources and codified business rules. | **Asset profiles** stored on the ledger to provide a comprehensive history in case of a claim. |
| **Fraud** declines precipitously thanks to transparent and immutable data on the ledger. | **Standards** for relevant claims data that are widely adopted among insurers and regulators. |
| **Expenses** due to loss adjustment become irrelevant as Blockchain transforms the insurance industry. | **A legal and regulatory framework** establishing the validity of smart contracts as binding instruments for insurance policies. |

*Table 4: Commercial property & casualty claims processing Card adapted from (Contri & Galaski, 2016)*

### Syndicated loans

Syndicated loans are the way to spread the risk of the very large amount borrowed by a client from more than one financial institution. This market could increase if their back-office operations were simpler. Nowadays, to select the syndicate members, a borrower needs to manually gather information from multiple sources. That means the underwriting and diligence systems doesn't communicate.

All this process creates delays and the need to intermediary parties resulting in increased costs. The Blockchain technology could provide the syndicated loan market with an easier, safer and more profitable solutions for financial institutions participations (Contri & Galaski, 2016).

#### Benefits

- Expedited: Smart contracts will automatically form syndicates, verify financial information and carry out settlement services, reducing the time to fund a borrower's loan;
- Abbreviated: Blockchain and smart contracts will eliminate the need for third-party intermediaries;
- Integrated: Diligence systems will communicate pertinent financial information directly to underwriting systems;
- Monitored: Regulators will have a real-time view of financial details throughout the syndicated loan lifecycle;
- Secure: Operational risk will decline as Blockchain automatically disburses principal and interest payments;

| Potential effects | Necessary conditions |
|---|---|
| **Syndicates** come together via smart contracts and under the watchful eye of regulators. | **A rating system** for counterparties that all financial institutions accept. |
| **Risk underwriting** requires substantially fewer resources to carry out effectively. | **Templates** for diligence and underwriting so that information can move from one system to another. |
| **Intermediaries** turn their attention elsewhere as smart contracts facilitate loan funding and servicing. | **The willingness** among financial institutions and loan requestors to store financial details on the distributed ledger. |

*Table 5: Syndicated loans Card adapted from (Contri & Galaski, 2016)*

### Trade finance

Trade finance is the bridge between exporters who need a guarantee of payment before they can ship and importers who need confirmations that the goods they paid will get delivered.

This process involves many risks, exporters use invoices to secure short-term financing from multiple banks, which increase the consequences if the delivery fails and parties use different platforms, increasing miscommunication, fraud and version control problems. All this process can result in invoices being financed more than once, delaying payments because of multiple checkpoints and slow the shipment of goods. The adoption of Blockchain could improve trade finance import/export efficiency by providing streamlined access to trade documents, greater capital efficiency and faster settlement (Contri & Galaski, 2016).

#### Benefits

- Accelerated: Time to shipment will shorten as financial documents are reviewed and approved in real time;
- Disintermediated: Banks facilitating trade finance will no longer require a trusted intermediary to assume risk or execute the contracts, eliminating the need for correspondent banks;
- Decentralized: Blockchain will show the status of contract terms are met, reducing the time and headcount required to monitor the delivery of goods;
- Trackable: Title and bills of lading available on the distributed ledger will show the location and ownership of goods;
- Visible: A real-time view of invoices and other essential documents will aid short-term financing, enforcement and AML.

| Potential effects | Necessary conditions |
|---|---|
| **Letters of credit** automatically generate from financial details stored on the ledger. | **Transparency** to ensure factoring and double spending isn't taking place. |
| **Regulators** gain real-time tools to enforce AML and customs-related activities. | **Interoperability** with legacy systems to accommodate letters of credit, bills of lading, and inspection documentation. |
| **Correspondent banks** exit the scene as import and export banks interact directly. | **Regulatory guidance** on the procedures that facilitate the use of smart contract reporting. |

*Table 6: Trade finance Card adapted from (Contri & Galaski, 2016)*

### Contingent convertible bonds

A Contingent Convertible (CoCo) bond is a hybrid security that combines features of equity and debt. This is a great opportunity for financial institutions to raise funds but have risks attached such as the market is largely untested, the instruments are highly volatile and there is a lack of insight.

By using Blockchain it is possible to create smart contracts that automate the regulatory reporting, minimizing point-in-time stress tests, reducing market volatility and improving investors confidence in these complex instruments.

#### Benefits

- Accessible: Confidence in CoCo bonds will rise as Blockchain provides up to date capital ratio information;
- Consistent: Standards will arise for the way banks calculate capital ratios and input them into the Blockchain;
- Immediate: Smart contracts will notify regulators of CoCo bond triggers as they happen;
- Compliant: With real-time insight into banks' capital ratios, regulators will have less need for stress tests;
- Responsive: Investors will claim their equity faster once the trigger condition is met;
- Sought after: A new rating system will encourage more institutional investors to participate in the market, raising demand for this type of bond.

| Potential effects | Necessary conditions |
|---|---|
| **Tokenized bond instruments** help investors make informed, data-driven decision. | **Standards**—including data fields, templates, trigger calculations and loan absorption—that apply across financial institutions. |
| **Smart contracts** alert regulators when loan absorption must be activated, minimizing the need for point-in-time stress tests. | **Processes** for regulators and bank leadership to act on real-time trigger notifications at the financial institution that issued the bond as well as across the market. |
| **Transparency** into loan absorption reduces the uncertainty of CoCo bond. | |

*Table 7: Contingent convertible bonds Card adapted from (Contri & Galaski, 2016)*

### Automated compliance

Compliance is a fact of life for banks. Audits, tax reporting, stress testing, and routine filing with the appropriate financial regulatory authorities are just a sample of what firms must do to stay in operation. All these operations have a cost. A Blockchain automated compliance market can eliminate error-prone manual work, reduce reporting costs and strengthen trust in their financial condition (Contri & Galaski, 2016).

#### Benefits

- Transparent: Data stored in financial systems will be immutable, accessible, and updated in real time;
- Painless: Automation will slash the time and resources required to perform an audit;
- Reliable: With permissioned access to financial data, audit teams will have a streamlined update process and avoid the errors that often arise from manual activities;
- Efficient: Reporting through the Blockchain will reduce duplicate effort and make it easier to prepare and file financial reports.

| Potential effects | Necessary conditions |
|---|---|
| **Audit software** dramatically reduces the time and resources required to examine accounts. | **Permissions** that allow each user to access only the financial data necessary to carry out their compliance responsibilities. |
| **Financial examiners** carry out their duties via permissioned access to pertinent financial information. | **Automatic enforcement** of compliance activity so that financial institutions and regulators share material information in real time. |
| **Costs** decline as the process for vetting transactions and filing reports becomes more straightforward. | **Interoperability** so that the legacy systems at financial institutions and regulatory agencies can communicate with the distributed ledger. |

*Table 8: Automated compliance Card adapted from (Contri & Galaski, 2016)*

### Proxy voting

Proxy voting is used when an investor can't attend a shareholder meeting. The company issues a proxy statement about the subject. A third party delivers the proxy to the investor and he delivers his decision to another outside party, who casts the ballot on their behalf. This process causes difficulties for shareholders. The proxy statements can contain errors, summaries can be misleading. And complicated, requiring intensive analysis to make an informed voting decision. Companies must mail print copies to any investor who doesn't elect to receive them electronically creating additional costs in the proxy distribution and in some markets, proxy statements can't be shared with institutional investors at all decreasing the potential number of votes. By using Blockchain to distribute proxy statements and count votes, it is possible to improve retail investor participation, automate validation of votes and enable personalized analyses (Contri & Galaski, 2016).

#### Benefits

- Direct: Storing all investment records on Blockchain will eliminate the need for a go-between to notify regulators and distribute proxy statements;
- Paperless: Costs from printing and mailing proxy statements will decline;

- Dependable: Smart contracts will ensure that voting is aligned to share ownership at the time of the vote;
- Accessible: Investors will have more ways (such as through mobile apps) to access proxy statements and cast their votes;
- Immediate: Depending on requirements, voting data will become available to the corporation and/or voters in real-time;
- Progressive: Evolving Blockchain applications will enable investors to conduct personalized, automated analyses.

| Potential effects | Necessary conditions |
|---|---|
| **Smart contracts** reduce the time and effort of distributing a proxy statement. | **Storage** of investment records on a distributed ledger to identify beneficial investors. |
| **Automatic reconciliation** prevents investors from casting more votes than the shares they own. | **Conversion** of votes cast via mail or phone into tokens to store on the distributed ledger. |
| **Self-service** enables investors to see vote counts and standardize analysis across investments. | **Collaboration** among corporations to develop a common voting solution. |

*Table 9: Proxy voting Card adapted from (Contri & Galaski, 2016)*

### Asset rehypothecation

Asset rehypothecation is a common practice in which a financial institution uses collaterals posted by its borrowers to cover trades of its own. This is also known as secondary trading. Rehypothecation reduces the cost of borrowing, but it can be tricky to manage. If the institution mixes up who owns what assets, the risk to trading partners goes up and the value of the rehypothecated assets becomes uncertain. The adoption of Blockchain could provide the asset rehypothecation market with a solution to remove much of the risk by automatically tracking assets and enabling real-time enforcement of regulatory control limits.

### Benefits

- Documented: Information such as collateral value, risk position and ownership history will be readily available to investors;
- Assessed: Counterparties will be rated based on transaction history, helping investors to hedge their risks;
- Automatic: Record-keeping, reporting and the movement of funds will take place without manual intervention;
- Observable: Regulators will have a clear view of the asset history, so they can enforce legal constraints;
- Orderly: Smart contracts will keep assets from being rehypothecated over regulatory limits;
- Stable: Between effective regulation and greater transparency, the risk of default leading to systematic failure plummets.

| Potential effects | Necessary conditions |
|---|---|
| **Ratings** based on prior transactions help counterparties make a better investment decision | **A tokenization standard** to represent collateral linked assets within the financial system. |
| **Reporting** of asset trades enables real-time enforcement of regulatory constraints. | **A common framework** for financial institutions to participate in the tokenized asset trading system. |
| **Controls** that terminate trades via smart contract technology reduce the likelihood of systemic failure. | **A Blockchain solution flexible** enough to handle changes in the over-the-counter (OTC) trading template. |

*Table 10: Asset rehypothecation Card adapted from (Contri & Galaski, 2016)*

### Equity post-trade

Equity post-trade is the process of swap trade details, change the record of ownership and exchange assets or securities between a buyer and a seller. Usually, this process takes from one to three days to complete and that’s for just one trade. A single market can deal with millions of trades every day. Adding the time, it takes, the dependence on costly intermediaries and the heavy regulatory compliance, the result is a huge expense on the equity post-trade market. By using Blockchain and smart contracts to post-trade activities it is possible to eliminate the need for intermediaries, reduce counterparty and operational risk and pave the way to faster settlement (Contri & Galaski, 2016).

### Benefits

- Swift: Same-day settlement will become a possibility thanks to automation and efficiencies like common data fields;
- Vetted: Automatic validation will strengthen custodians' confidence that a counterparty is able to settle;
- Connected: Investors will receive immediate notification of trade settlement without relying on a custodian;
- Straightforward: When securities settlement systems become unnecessary, custodians will have more say in how to store assets;
- Empowered: Servicing activities initiated via smart contract will eliminate the need for third-party intermediaries;
- Wrinkle-free: Technology and manual errors will decline when smart contracts transfer equity and cash.

| **Potential effects** | **Necessary conditions** |
|---|---|
| **Automation** of post-trade processes reduces settlement time and lowers counterparty risk. | **Incorporatio**n of 'net transaction' benefits within settlement in order to minimize transfers across custodian banks. |
| **Smart contracts** simultaneously transfer equity and cash in real time, reducing the likelihood of errors. | **Collaboration** among regulators, custodians and exchanges to develop a solution that can provide market stability while serving everyone. |
| **Disintermediation** of clearing, settlement and asset servicing reduces operational costs and third-party fees. | **Standardization** of data fields that can match trades while preserving investor confidence and anonymity. |

*Table 11: Equity post-trade Card adapted from (Contri & Galaski, 2016)*

### Global Payments

Global payments mean transferring money across international borders. To regulate this transaction, it is required the intervention of a third party, usually a bank or a money transfer operator (MTO). This operation takes time and it is expensive. Using Blockchain technology in global payments will enable lower fees, real-time settlement and newer models of regulatory oversight (Contri & Galaski, 2016).

#### Benefits

- Incorruptible: Digital profiles stored on Blockchain will authenticate both sender and beneficiary;
- Liquid: Through smart contracts, participants willing and able to convert fiat currencies will support the foreign exchange;
- Prompt: Cross-border payments will be completed in real time;
- Economical: With fewer participants, improved cost structure can generate value;
- Visible: regulators will gain automatic alerts to specific conditions along with on-demand access to complete transaction histories over the ledger.

| Potential effects | Necessary conditions |
|---|---|
| **Real-time settlement** of international money transfers reduces liquidity and operational cost. | **KYC standards** that are consistent across banks and MTOs. |
| **Direct interaction** between sender and beneficiary banks eliminates the role of the correspondent bank. | **Binding legality** of a Blockchain-enabled global payments solution. |
| **Trust** improves as smart contracts capture obligations across financial institutions. | **The consensus** among financial institutions around Blockchain platforms and adoption timetables. |

*Table 12: Global Payments Card adapted from (Contri & Galaski, 2016)*

## 3.2. Retail and Consumer Goods

Retail and consumer goods are one of the many markets that can beneficiate with Blockchain technology. Providing the retail and consumer goods industries with an opportunity for trustless transactions may sound counterintuitive, but in doing so it engenders consumer trust. Based on the following main use cases of this market, it is analysed and evaluated the advantages that Blockchain technology can provide ("Blockchain Research Lab," n.d.).

### Smart tags

Smart tags are attached to or implemented in a physical product. Their function is to hold information about the product and ownership reference. Through Blockchain technology, it can be provided a new way for entities to register and sell, transparently and permanently, any asset.

#### Benefits

- Verify the Authenticity of Products and Create a Trustworthy Secondary Market;
- With smart tags, producers prevent faked products to be sold illegally (black market) and impose a significant hurdle for counterfeiters to imitate their products;
- Smart tags impose a hurdle for reselling stolen products;
- Accessible and verified information about products potentially increases resale prices;
- Consumer protection against buying stolen or faked products in good faith;
- Disintermediation – e.g. artists and musicians are not obligated to market their work via the use of third parties;
- Blockchain provides a transparent register;
- Reduction of criminal activities, like theft and counterfeit of digital goods.

### Digital Goods: Micropayment and Pay-per-Use

Artists, musicians or publishers can introduce new payment schemes and customers can pay only for the products they have actually used.

### Benefits

- Disintermediation;
- Transparency;
- Blockchain enables micropayments.

### Improved Supply Chain Management

Using Blockchain products can be monitored and displayed to enable transparency and transaction data.

### Benefits

- Improved authority supervision;
- Improved product safety and transparency;
- Secured Oversea shipments through Blockchain-based trade finance solutions;
- Comprehensible distribution and processing of contaminated goods in case of product recalls.

### Sharing Economy: Borrowing - Hiring - Sharing

Blockchain will create a safe and fast environment for monetary transactions, verifying and changing ownership references, and recording usage behaviour. This can result in a new kind of sharing economy.

### Benefits

- Transparent ownership references;
- Implementation of quick and safe payment schemes;
- Usage-behaviour of participants/owners are recorded and respective payment schemes can be implemented;
- Investments and maintenance are transparent and accountable;
- New revenue streams for producers: offering installation, repairs, maintenance, add-on features, and bringing together consumers to share;
- Shared costs.

### B2C Certificates

The consumer will be able to check the validity of proclaimed product features using Blockchain-based information.

### Benefits

- The consumer can verify instantly at the PoS;
- Data is reliable due to the maintenance of authorities;
- Adjustments are immediately online;
- Consumer protection against falsely claimed product features;
- The consumer is informed about low-quality certificates and labels.

## Consumer Participation in Product Development

If consumer and producer interact during the development phase of new products, consumers could buy Blockchain-based shares of products and influence decision-making processes, the producer thereby could increase the quality of product development and consumer feedback

### Benefits

- Alternatives to traditional funding instruments for producers;
- Creating communities around a brand and/or product;
- In order to increase the value of their participation, (micro-)shareholders are incentivized to advertise brands/products;
- Increased quality of valid consumer feedback and potentially shortened product development cycles.

## Product Warranties and Insurance

Creating an appealing application for digitally storing product warranties and insurances for multiple devices allows consumers to adequately manage and survey claims in case of damage or defect.

### Benefits

- Allows for fast handling of warranty and insurance cases, as information regarding the product and regarding changes to the product are safely stored on a Blockchain and thereby transparent to the involved parties;
- In the case of a resale by the consumer, ownership references can be adjusted safely and quickly;
- In the case of a resale, the trustworthy information about warranties and the condition of the product might increase resale-price and hamper fraudulent behaviour;

- For producers, such application enables to create new revenue streams, recommending repair-services, insurances and add-on features;
- Referring to the IoT (Internet of Things), products like Washing Machines, Mobile Phones, etc., can communicate usage-related information, consumer-initiated repairs and their respective consequences on the validity of warranties and insurances.

### Verification, Management and Selectively Forwarding of Personal Documents

Documents can be registered, managed and forwarded via the Blockchain. Algorithmic hashing allows encrypting regarded documents, empowering the individual user to reveal it selectively to parties. Thereby the Blockchain is utilized as the trustworthy container of verified data.

#### Benefits

- Streamlined verification of identities and processing of online purchase orders;
- Data integrity.

### Incentivizing Online WoM (Word of Mouth)

Producers can utilize Blockchain and tokens to incentivize online WoM and track referrals and link transactions.

#### Benefits

- Tokens impose incentives and/or payment;
- Transparency of transaction history;
- Increased customer loyalty;
- Increased WoM (hence potentially increased acquisition rates).

### Automated In-Store Payments

By combining Blockchain-based payments and smart tags on Fast-moving consumer goods (FMCG) products, it will allow a better shopping experience as well as cost-cutting.

### Benefits

- Immediate Payment;
- Cost-cutting through less required staff;
- Cost-cutting due to lower costs of payment processing.

### Cost-Cutting: Direct FMCG Purchases by Consumers

With payments and customer loyalty programs on Blockchain, consumers can avoid paying retailer's margins by buying directly from the producer.

### Benefits

- Cost cutting by dis-intermediating the middlemen;
- Anonymized purchases by the consumer, thereby linking purchase behaviour and preferences to the anonymized profile (adjustable by the consumer);
- Producers can advertise and recommend products to the anonymized profile, based on its preferences and buying behaviour;
- New revenue streams for producers, offering repair services, add-on features, personalized offerings based on preferences;
- The consumer is in control of his data and can selectively forward it;
- Involving customer loyalty programs, producers can intensify customer relationships.

### Product Quality and Food Safety

Blockchain imposes a secure and fast infrastructure for comprehensively communicating, verifying and adjusting company-related information. This scenario can be extended to authorities observing deliveries between companies and manufacturers creating an observable supply chain, in search of dangerous or contaminated production and more effectively organize product recalls.

### Benefits

- Coordination and combination of numerous reliable sources of information;
- Validity of data;
- Security of data;
- Quick communication of adjustments to a company's status;
- Comprehensive evaluation of company compliance.

### Decentralized Marketplaces

Via decentralized marketplaces individuals can induce trades directly between seller and buyer, eliminating third parties and related costs.

Benefits

- Huge cost-cutting potential;
- Disintermediation;
- Anonymized shopping.

### Customer Loyalty Programs

Blockchains present an opportunity for producers to link product purchases with customer loyalty programs, utilizing tokens which are addressed to accounts on Blockchain and can be used or accumulated by consumers.

Benefits

- Immediate payment;
- Incentivizing customer loyalty and re-purchases through token payment;
- Eased processing by linking payment and token transactions;
- Personalized offerings.

## 3.3. Government and Democracy

Blockchain's influence in the public sector will be mostly behind the scenes. But the technology has the potential to bring security, efficiency, and speed to a wide range of services and processes, that's why so many government leaders are actively exploring its uses in government. Indeed, Blockchain experiments in the public sector are accelerating.

It is the Government's responsibility to create an incentive for ensuring accurate transfers of value between relevant stakeholders. Across those many transactions and business events, numerous use cases for Blockchain present themselves

“For the purposes of considering where Blockchain could likely be adopted within and across government, three business values of Blockchain—recordkeeping, value transfer, and smart contracts—provide broad cases for possible adoption” (White, Killmeyer, & Chew, 2017). Those three business value are presented and exemplified on table 13.

| | Business Value | Examples |
|---|---|---|
| *Value transfer* | Low-cost and near real-time.<br>Without an intermediary.<br>Beyond money. | Domestic and international remittance.<br>Internal payments settlement.<br>Clearing and settlement of securities.<br>Exchange of low liquidity assets. |
| *Smart contracts* | Software protocols.<br>Based on ledger content.<br>Execute when the conditions are met. | Digital cheques/IOUs.<br>Automatic financial instruments.<br>Parametric insurance contracts.<br>Automated market making. |
| *Recordkeeping* | Create immutable record.<br>Under agreed consensus protocol.<br>Without reliance on a trusted third party. | The digital certificate of ownership for physical assets.<br>Transaction validation of digital assets.<br>Financial accounts. |

*Table 13: Three primary areas of Blockchain delivers business value (White et al., 2017)*

Next, it will be present some use cases where governments could leverage by implementing Blockchain. Such use cases are identity management, land registration, and voting. For each one it will be described the current pain points and how could Blockchain improve them (White et al., 2017).

### Identity management

Digital identity is both a use case for Blockchain and the enabler that allows any other assets to be transacted on the Blockchain. In order to be transacted, each asset needs to be digitalized, and the owner or transactor also needs a digital identity that proves its veracity to engage in those transactions. This is a challenge recognized by governments around the world because there is a considerable portion of the world's population that lives without a legal or officially recognized identity.

Existing pain points:

- Lack of standards for establishing a digital identity;
- Differing attestation processes and identity "entry points" prevent economic engagement and can hinder public sector service provision.

Blockchain value proposition:

- A secure, self-sovereign identity could enable efficient transactions across a wide variety of asset classes;
- Individual and explicit control over which identity elements are shared for which purposes.

### Land registration

By securing a unique and non-corruptible record on a Blockchain and validating changes to the status of that record across owners, it is possible to create a reliable property record. Taking into account that these records not only provide critical protection for homebuyers in developed nations but also serve as a basis for investment and economic growth across many developing nations. According to White, Killmeyer, and Chew (2017), governments are exploring Blockchain in land registration on every continent but Antarctica.

Existing pain points:

- License and registry processes are paper-based and fragmented, making transactions costly, inefficient, and vulnerable to tampering;
- In the United States, landowners spent 800 million dollars in 2014 and 2015 on title insurance to cover risks associated with real estate titles.

Blockchain value proposition:

- A decentralized, standardized system for land registration records could reduce the number of intermediaries required, increase trust in the identity of transacting parties, increase process efficiencies, and decrease time and cost to the process;
- Recording property rights via Blockchain would enable 2 to 4 billion dollars in annual cost savings in the United States alone for title insurers through a tamper-proof ledger.

### Voting

There have been many Blockchain projects working to provide secure digital identity management, anonymous vote-casting, individualized ballot processes and ballot casting confirmation verifiable by the voter. The expected result is a solution that will enable citizens to cast votes the same way they initiate other secure transactions and validate that their votes were cast or even verify the election results.

Existing pain points:

- High costs related to ballot printing, electronic voting machines, maintenance, etc;
- Increasing threats of cyber-attacks compromising election results;
- Lack of transparency due to a centralized process of election results audit;
- Voting delays or inefficiencies related to remote/absentee voting.

Blockchain value proposition:

- Cost savings through Blockchain-enabled voting;
- Enhanced security and audibility of votes;
- Greater participation in elections, including remotely;
- Greater transparency meeting citizens needs.

## 3.4. Military

The recently increased number of military Blockchain projects makes it obvious the growing interest of military agencies in the Blockchain technology and its potential for distributed, resilient and tamper-proof identity systems.

Bitcoin Magazine reported that both the Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense (DoD) and NATO have requested proposals for the development of military-related apps built on Blockchain technology.

According to Giulio Prisco, 2016, DARPA wants to leverage Blockchain technology to create a secure messaging service and NATO is interested in applications of Blockchain technology to military logistics, procurement and finance. Previously, the U.S. Air Force worked with contractors to develop a Bitcoin payment gateway (Prisco, 2016).

A technology such as a Blockchain if it can be validated to be able to support the appropriate level of security and privacy, has potential applicability to multiple information sharing use cases within the homeland security enterprise.

The Celerity Government Solutions, a security provider organization, is researching Blockchain solutions to enable users to establish and maintain trusted identity transactions with public and private organizations. The project is titled "Blockchain Software to Prove Integrity of Captured Data From Border Devices," and it will create an identity log that captures a list of a device characteristics while adding the dimension of time for increased security.

Their objective is to limit hacker's abilities to corrupt the past records for a device, making it more difficult to spoof. NATO request for proposal also included an IoT section, which underlines the synergy between IoT and Blockchain technologies for military applications

### 3.5. Energy and Power

The Energy and Power sector can also be improved using Blockchain technology. Making possible the trading of energy between users of the network infrastructure. On figure 10, it is presented some of the possible use cases in the energy sector that can be remodelled by Blockchain.

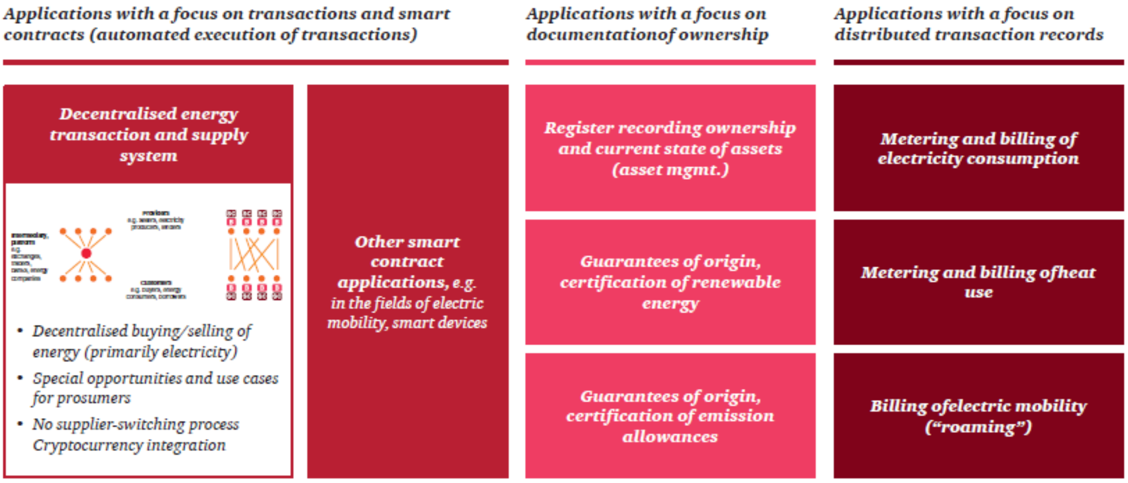

Applications with a focus on transactions and smart contracts (automated execution of transactions)
Applications with a focus on documentationof ownership
Applications with a focus on distributed transaction records
Decentralised energy transaction and supply system
Decentralised buying/selling of energy (primarily electricity)
Special opportunities and use cases for prosumers
No supplier-switching process Cryptocurrency integration
Other smart contract applications, e.g. in the fields of electric mobility, smart devices
Register recording ownership and current state of assets (asset mgmt.)
Guarantees of origin, certification of renewable energy
Guarantees of origin, certification of emission allowances
Metering and billing of electricity consumption
Metering and billing ofheat use
Billing ofelectric mobility ("roaming")

*Figure 9: Overview of possible Blockchain use cases in the energy sector (Hasse et al., 2016)*

### Decentralised energy transaction and supply system

The Blockchain technology could implement a decentralized energy supply and radically simplify today's multi-tiered system, in which power producers, transmission system operators, distribution system operators and suppliers could exchange transactions directly with each other. This way it is possible to connect producers with consumers and eliminate the need for third parties from the whole operation (Hasse et al., 2016).

It is possible for the energy distribution system to be controlled through smart contracts, by deciding when to initiate certain transactions based on predefined rules. Using this method will ensure that all energy and storage flows are controlled automatically to balance supply and demand. Balancing activities and virtual power plants could be managed by smart contracts using decentralised storage of all transaction data on a Blockchain making it possible to keep a distributed, secure record of all energy flows and business activities.

Another potential future area of application could be using cryptocurrencies to pay for the energy supplied or using Blockchains for documenting ownership and related transactions, by providing secure storage of ownership records (Hasse et al., 2016).

By using an energy process to exchange energy on a Blockchain, it would no longer require energy companies or other third parties, instead, a decentralised energy transaction and supply system would emerge, using smart contract applications and empower consumers to manage their own electricity supply contracts and consumption data.

There are many applications that could be accomplished using a Blockchain-based model, such as simplifying bills at charging stations, which may be located in public spaces where they can be used by anyone, using, for instance, cryptocurrencies paying methods that would widen the use of electric vehicles.

### Benefits

- Lower transaction costs due to the cutting out of intermediaries;
- Falling prices as a result of greater market transparency;
- A simple option for customers to become a service/electricity provider;
- Transactions are generally made simpler (documentation, contracts, payment);
- Greater transparency thanks to decentralised data storage;
- Flexible products (tariffs) and supplier switching;
- Strengthening of prosumers thanks to independence from a central authority (direct purchases/sales of energy).

### Risks

- Complete loss of data on loss of ID;
- Currently high transaction costs for public Blockchain systems;
- Possibly lack of acceptance on the part of consumers;
- No authority in the case of disputes, no direct possibility of escalating conflicts;
- Risk of fraudulent activities at the interface between the real world and the digital Blockchain world;
- Lack of long-term experience;
- Technical problems with initial applications possible to start with;
- Insufficient or inadequate functionality and security risks due to lack standardisation;
- Networks must cope with greater flexibility.

Some examples of projects being made on this field are RWE energy company in collaboration with Slock.it are working on BlockCharge, a Blockchain application for charging electric vehicles. The Vattenfall energy company has a project named Powerpeers that is a Blockchain marketing of electricity. The LO3 Energy is developing Blockchain based innovations to revolutionize how energy can be generated, stored, bought, sold and used, all at the local level. Like these, there are many other companies working on similar projects.

## 3.6. Higher Education

The Blockchain technology can provide a global network platform for higher learning that could be rich in information, secure and transparent.

The opportunities for this sector can be separated into four categories:

- Identity and Student Records;
- New Pedagogy;
- Costs (Student Debt);
- The Meta-University.

### Blockchain, Identity, and Student Records

In order to make possible the adoption of Blockchain in higher education, it is required to overcome a few major challenges (Tapscott & Tapscott, 2017).

The first challenge is to maintain the privacy and security of data stored digitally by those academically accredited institutions, the Blockchain properties can assure that all data would be tamper-proof and at the same time, the privacy of the users would not be violated. The way to achieve this goal is by inserting in plaintext the information that the student and the educational institution and apply the public key encryption system to the student's personal data.

A second challenge to address is validity. It is possible to solve this challenge by providing a reliable, cost-free and fast trade of information that would beneficiate employers, creating a tool to search and verify the candidate's information, but also students that would have free access to their information and share it with employers.

"According to CareerBuilder, 57 per cent of job applicants have embellished their skill set, and 33 per cent have lied about their academic degree. Not surprising, employers are wanting to see official college transcripts. However, when it comes to processing requests, universities often charge transaction fees" (Tapscott & Tapscott, 2017).

A third challenge is a time. The MIT Media Lab started hashing digital certificates onto the Blockchain to permanently denote membership and to reward community members for their valuable contributions to the lab's work. Like the MIT, other Blockchain projects could be developed to exploring ways to reward students with credentials for everything they learn, no matter the setting. Students would no longer just getting a grade but a credential instead, which they could put to use immediately on the job market.

## Blockchain and the New Pedagogy

One of the main aspects of higher education that should rethink is the model pedagogy. Big universities are still offering the broadcast model of learning, in which the teacher is the broadcaster and the student is the supposedly willing recipient of the one-way message. It is true that colleges and universities are working hard to move beyond this model, but unfortunately, it remains dominant.

The purpose of higher education should also be rethought. It is not about skills, and to a certain extent, it is not even about knowledge. What counts these days is the capacity to learn throughout life; to research, analyse, synthesize, contextualize, and critically evaluate information; to apply research in solving problems, and to collaborate and communicate.

One of the first Ethereum software-development companies tried to find a solution to this problem creating the "Consensus Systems (ConsenSys)" (Tapscott & Tapscott, 2017), their purpose was making members of ConsenSys choose two to five projects to work on. There are no top-down assignments or a boss, everyone owns a piece of every project directly or indirectly. The goal is to achieve a balance between independence and interdependence, the students will improve their agility, openness, and consensus by identifying what needs to be learned, distributing the load among the students eager and able to do it, agreeing on their roles, responsibilities, and rewards, and then codify these rights in smart contracts.

## Blockchain and Costs (Student Debt)

The education sector can be seen as a business considering the number of companies that make their fortunes providing classroom content, additional teacher training, classroom and school administration systems, and the testing content and platforms. All this can be extremely expensive considering the fees that students have to pay to have access to education. There are projects being developed based on Blockchain in order to solve this problem.

"Melanie Swan is looking to the Blockchain to tackle student debt head-on. She is the founder of the Institute for Blockchain Studies. She has been working on MOOC accreditation and "pay for success" models on the Blockchain" (Tapscott & Tapscott, 2017)

There are three elements toward this goal and all can be achieved using Blockchain. It is required a trustable proof-of-truth mechanism to confirm that the students who signed up for Coursera classes and actually completed them, a payment mechanism, and the last one, smart contracts that could constitute learning plans.

“The visionaries behind the Learning Is Earning initiative, such as Jane McGonigal, in partnership with the Institute for the Future and the ACT Foundation, envision “teach it forward” schemes in which students can pay down their student loans by teaching other students what they just learned or by applying this new knowledge immediately in the job market” (Tapscott & Tapscott, 2017).

In this project employers, students or professors will be able to use the Blockchain to search for people with the particular combination of skills and knowledge needed immediately on the job or in the classroom, helping employers match projects with the proven capabilities of students available for project work.

### The Blockchain and the Meta-University

The Blockchain will enable the 21st-century institution of higher education to disaggregate into a network and an ecosystem. Indeed, innovators have an enormous opportunity to create an unparalleled educational experience for students globally by assembling the world’s best learning materials online and enabling students to customize their learning path with support from a network of instructors and educational facilitators, some of whom may be local and some halfway around the globe.

To make this work for students, colleges and universities will require deep structural changes, and educators will need to embrace partnerships.

One of the main goals of using the Blockchain in higher education is to create a rich, secure and transparent network for higher learning. To achieve this goal three stages were envisioned.

The first stage is a content exchange, where professors could share ideas and upload their teaching materials to the Internet for others to use freely. The second one is content co-innovation, where teachers collaborate across institutional and disciplinary boundaries to co-create new teaching materials using wikis and other tools. Finally the stage three, where the college or university would become a node in the global network of faculty, students, and institutions. Creating a platform for learning collaboratively (Tapscott & Tapscott, 2017).

## 3.7. Manufacturing

Although the majority of research projects are still focused on the technology itself and applications in the finance industry, the interest to exploit Blockchain in the manufacturing industry is increasing. Especially the application of Blockchain for supply chain management and auditing is investigated by several start-ups and large companies.

| *Use case* | Examples | Description |
|---|---|---|
| *Supply Chain Management and Digital Product Memory* | IBM and Maersk.<br>Provenance.<br>Everledger. | Tracking of containers during the shipping process.<br>Recording of all important product information throughout the entire supply chain.<br>Registers certifications and transaction history of diamonds on the Blockchain. |
| *Internet of Things and Industry 4.0 applications* | Factom Iris.<br>Super Computing Systems.<br>Tile Data Processing tilepay.<br>IOTA.<br>IBM Watson IoT. | IoT device identification over Blockchain.<br>Sensors that timestamp data on the Blockchain to save them from manipulation.<br>Marketplace to allow customers to sell their data from IoT devices.<br>Cryptocurrency and Blockchain protocol especially developed to meet the demands for IoT applications.<br>Platform to save selected IoT data on a private Blockchain and share it with all involved business partners. |
| *3D printing* | Genesis of Things.<br>Moog Aircraft Group. | Platform to enable 3D printing via smart contracts<br>Ensuring safe 3D-printing of aircraft parts via Blockchain |

*Table 14: Overview of Blockchain use cases in the manufacturing industry adapted from (Dieterich et al., 2017)*

### Supply chain management and digital product memory

IBM and Maersk, the leading shipping company, tested the application of Blockchain in logistics, where they prove that a Blockchain can be used to track containers during the shipping process (Dieterich et al., 2017).

Their goal was to reduce the effort and paperwork that is necessary for the shipment. All actors in the supply chain can access the information that is relevant to them using the platform and they can act on it. By reducing the paperwork, IBM and Maersk hope to reduce the shipment costs dramatically by providing information more rapidly and preventing shipping fraud.

Project Provenance Ltd is a start-up which is trying to secure the traceability of certifications and other important information about products on a Blockchain. Their idea is that every product gets a "digital passport" that proves its authenticity and helps to determine its origin, thereby preventing the sale of fake goods (Dieterich et al., 2017).

Using Blockchain to register every step of the production process it is possible ensure that the transfers of ownership are explicitly authorized by their relevant regulators without having to trust the behaviour or competence of a third entity. They create different software to better meet the necessities of each participant, considering what is the relevant information for each one. Afterwards, the buyer can scan the product and access the information from the Blockchain to check every step of the production process.

Everledger is trying to increase the trust in products using Blockchain to register diamonds and secure their transaction history and ownership. "In the future, the start-up wants to extend the application of their technology to more luxury goods. In addition, the CEO of Everledger believes that the technology can also be beneficial to identify machines in an IoT context" (Dieterich et al., 2017).

### Internet of Things and Industry 4.0 applications

"Factom Irisy realized that the current form of authentication based on certificates from authorities is too expensive for the IoT and that the scalability is questionable" (Dieterich et al., 2017). Their goal is to register the devices on a Blockchain to create a digital identity of the device which cannot be manipulated. Offering the advantage of dynamically add and update information about the device.

The Super Computing Systems AG published a white paper proposing the usage of Blockchain to timestamp sensor data for Industry 4.0 applications. Their purpose is to increase the level of trust between different parties, creating sensors that can save and thereby timestamp their data on a Blockchain. Ensuring this way that the data was not manipulated afterwards and that all standards were met (Dieterich et al., 2017).

Tile Data Processing Inc. investigates the usage of Blockchain to provide access to data that is generated by IoT devices. Ideally, customers will be able to sell their IoT data via the service "tilepay", where they can register and collect their data and decide who can purchase it. Data can be purchased in real-time and companies can make a direct peer-to-peer payment to the customer via Bitcoin. Limited scalability, low verification speed and incurring transaction fees are several technical challenges that need to be solved (Dieterich et al., 2017).

IOTA, a cryptocurrency especially built for the IoT, solves those problems by using a different kind of algorithm. Instead of using a classic Blockchain, a directed acyclic graph called tangle is used. Their solution forces every participant who wants to make a transaction to first approve two previous ones. In the case of conflicting transactions, a tip selective algorithm is used and the more likely one is chosen. This new algorithm increases the verification speed and allows better scalability, and at the same time helps machine-to-machine microtransaction to become economically reasonable (Dieterich et al., 2017).

IBM introduced their Watson IoT platform that helps companies to save selected IoT data to a private Blockchain, which is used to share the protected data among all business partners involved. This project enables small and midsized companies to leverage the benefits of IoT, by developing a platform for supply chain, trade lane, asset management, regulatory and compliance use cases.

### 3D printing platforms

A working paper published by Blechschmidt and Stöker (2016) approach how Blockchain can eliminate the overhead in the manufacturing industry, which they call the "trust tax". The goal of this project is to create a platform based on Blockchain to facilitate the 3D printing supply chain. First, the designer registers his product design encrypted on the Blockchain. Then the design file uses smart contracts to automatically negotiate to price, find the nearest and cheapest 3D printer and negotiate conditions with the customer and the logistics service provider (Dieterich et al., 2017).

This knowledge can not only increase the trust of the customers, by including the entire product history, but also enables large cost savings when it comes to warranty, maintenance and excluding the need for a middleman.

Moog Aircraft Group is running a project where the Blockchain is used to securely transfer the data to a verified 3D printer. "They want to use 3D printing to enable a point-of-use and time-of-use supply chain, where aircraft parts can be printed exactly when they are needed, saving inventory, import and logistic costs" (Dieterich et al., 2017).

A scan of the grain structure of each part is used as a fingerprint, guaranteeing that each individual part can be identified without a doubt, helping technicians to ensure that it was not counterfeit before the installation into an aircraft. In addition, they want to use the platform to produce spare parts for discontinued aircraft models.

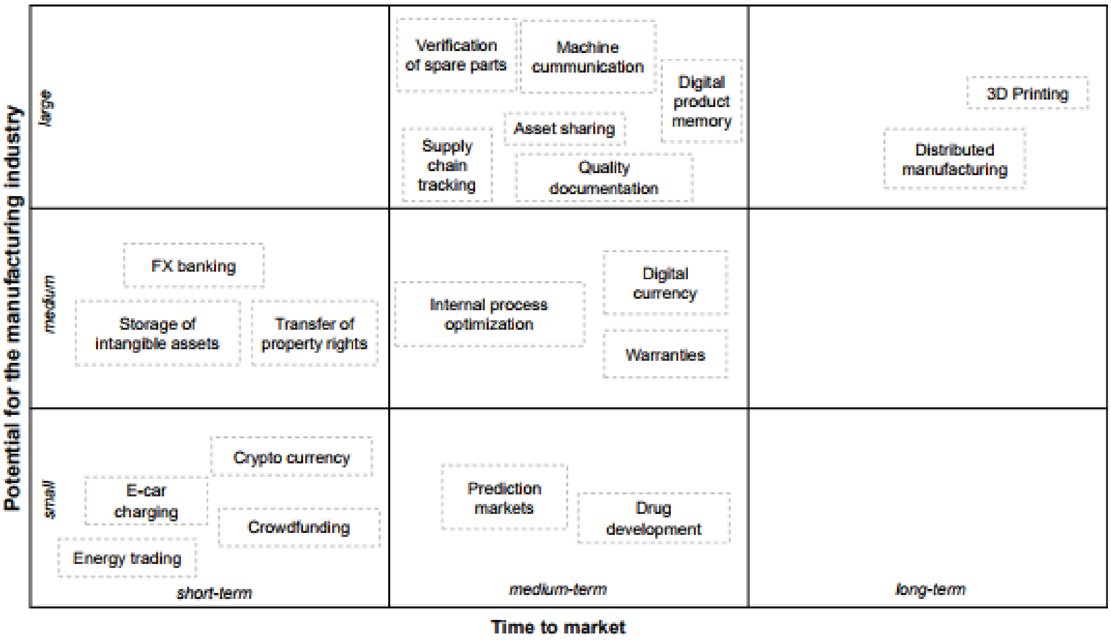


*Figure 10: Blockchain use cases and potential application for manufacturing (Dieterich et al., 2017)*

On figure 11, it is categorized a list of use cases of the manufacturing industry by crossing them with two dimensions.

The first one is "time to market" because currently, most use cases are only proofs of concept and still have to master the market entry stage. The second dimension is the "potential for the manufacturing industry". Currently, there are not many applications specifically in the manufacturing industry and these also have different potentials for a significant impact.

The objective is to show the high potential of Blockchain for the manufacturing and machine tool industry. Based on expert interviews and a market survey, a variety of use cases of Blockchain technology in the manufacturing industry was identified and analysed using cluster analysis and evaluated based on criteria for a beneficial application of Blockchain.

## 3.8. Media and Telecommunications

Having learned about Blockchain technology in general and its particular relevance for the media industry. There are five Blockchain-based use cases, figure 12, that are considered the most relevant for this industry (Deloitte, 2017).

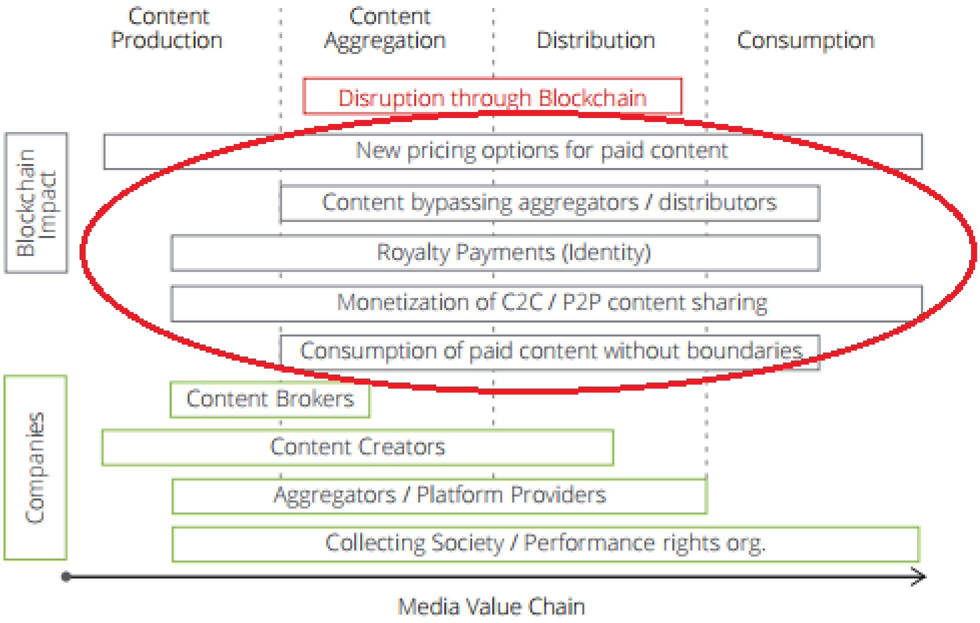

Content Production
Content Aggregation
Distribution
Consumption
Disruption through Blockchain
Blockchain Impact
New pricing options for paid content
Content bypassing aggregators / distributors
Royalty Payments (Identity)
Monetization of C2C / P2P content sharing
Consumption of paid content without boundaries
Companies
Content Brokers
Content Creators
Aggregators / Platform Providers
Collecting Society / Performance rights org.
Media Value Chain

*Figure 11: Blockchain's primary relevance in the media value chain adapted (Deloitte, 2017)*

### New pricing options for paid content

Consumers expect "per-use" payment models, instead of paying a monthly/yearly fee for an online subscription to one particular newspaper/(Pay-)TV channel.

Blockchain can enable micro-payments and help publishers to monetize this flexibility seeking a group of customers. With the help of a Blockchain, individual articles or other pieces of content could be sold at a lower cost without disproportionate transaction costs.

#### Benefits

- Increased willingness to pay especially younger digital natives are more willing to pay a few cents for a music track they favour than to be charged a flat monthly subscription;

- Copyright tracking becomes more accurate, as does allocation to media copyright holders and the subsequent distribution of royalty payments;
- Efficiency increases, since costly monitoring of contractual agreements and complex distribution of profits are not necessary.

### Challenges

- Transaction quantity is massive because a large quantity of historical data needs to be retained at the Blockchain nodes, due to the number of transactions;
- Common Blockchain standards still need to be agreed on;
- Initial user registration is inevitable. Users have to register and provide payment details to activate pay-per-click;

## Content bypassing aggregators

The digital adverting ecosystem still remains important in the next decade. It is complex and involves numerous stakeholders. There are several intermediaries between the content creator and the potential advertiser. The slice of the monetization cake for the initial creator of digital content becomes smaller with every additional party involved.

Based on the Blockchain, everyone from leading media houses to small bloggers can easily generate advertising revenues. As Blockchains permit an exact tracking of content usage, it also enables a direct allocation of advertising budgets.

### Benefits

- Blockchain permits direct customer relationships between fans and artists;
- Marketing performance and impact become more accurately measurable;
- Existing complex media and advertising ecosystems become simple and transparent.

### Challenges

- Content aggregators and advertising networks are likely to lose their dominant market position in the media world;
- Monetization of content becomes more democratic and entry hurdles could vanish.

### Distribution of royalty payments

Today, the distribution of royalty payments builds on multiple contracts between artists, producers, and music publishing houses. To ensure that this is happening, the national copyright collection bodies act as a collection platform for copyright holders and compensate the eligible parties.

However, contractual complexities can complicate the settlement activities, leading to opaque proceeds. The share of royalty payments distributed in this manner relates to music consumption that cannot be linked to the rights holder.

With the help of a Blockchain, the distribution of royalties could become more efficient and transparent. This would include a music directory with the original digital music file associated with all relevant identities of people involved in the content creation. It is also possible to store instructions in form of smart contracts that specify how the artists are to be compensated and how sales proceeds are to be divided among all eligible parties.

#### Benefits

- Near real-time and exact allocation and distribution of royalty payments according to usage, based on smart contracts;
- Cost efficiency, no costly tracking and monitoring systems for music usage required, as every consumption/usage will be tracked in the Blockchain;
- New role of collection associations, Blockchain platform provider and verification of smart contract details through collection associations as trusted third parties.

#### Challenges

- Large amounts of historical data to be retained at the Blockchain nodes due to the number of “transactions” (airplays, streams, club-rotations etc.) across all music consumption channels;
- Common Blockchain platform and interoperable Blockchain standards need to be agreed upon by the many relevant participants;
- The position of a trusted third party might not be granted to collection associations by market participants.

### Secure and transparent C2C sales

Blockchain has the potential for content rights owners to enable additional revenue streams by leveraging consumer-to-consumer sales. Peer-to-peer networks and the respective exchange of (media) files is almost impossible to control due to the sheer number of exchanges and of users exchanging files. Attempts to legalize file exchanges and to monetize the transactions and contents have failed, owing to lack of customer interest and acceptance.

Nevertheless, illegal file sharing remains a major problem for media companies, while the Blockchain has the potential to solve that problem.

With a Blockchain, content owners have full control and visibility of the consumption and number of uses of individual songs and/or movies. Therefore, piracy and copyright infringements are nearly impossible. In addition, the transparency of Blockchain enables content owners to “control” peer-to-peer content distribution and thus to create new business models such as consumer-to-consumer marketing of content.

### Benefits

- Content owners can fully leverage, control, and monetize all copyright assets that are recorded in the Blockchain. In addition, illegal file sharing and other copyright infringements will be impossible, due to the transparency of the Blockchain details through collection associations as trusted third parties;
- The Blockchain records every usage of a specified content and enables real-time and fully transparent consumption-based pricing mechanisms. Consumers do not have to pay a monthly up-front fee, instead, only the actual usage will be billed to consumers;
- Due to the very low transaction costs in the Blockchain, consumption-based business models are also applicable to micropayments.

### Challenges

- Media aggregators will still play a role in the marketing of contents. Nevertheless, it is expected the dynamics of the market to change in the long run due to the “democratizing” effect of Blockchain. The aggregator role will shift towards curated discovery platforms to find new content and will lose their “gate-keeper” role, as monetization and real-time billing will be available to content owners via Blockchain.

### Consumption of paid content without boundaries

The last use case deals with a situation that many subscribers of paid content subscriptions have witnessed in the past. They cannot access the contents they subscribed to once they are in another country/region, for example during business travel or on vacation.

Blockchain has the potential to make Digital Rights Management (DRM) systems obsolete or at least to reduce the complexity of these systems because every transaction/consumption is tracked in the Blockchain and directly linked to a user. The payment will be automatically initiated according to the underlying smart contract terms for the content.

### Benefits

- Improved customer experience through “seamless” subscription models across different geographic areas;
- Less complex and real-time billing;
- Transparent and “self-executing” rights management due to underlying smart contract.

### Challenges

- Transformation from currently installed DRM and billing systems towards multi-country access and integration of Blockchain functionalities is fraught with complexities;
- Players could become obsolete as aggregators since content owners will have the ability to market and sell their intellectual property directly to consumers.

In a nutshell, Blockchain’s potential benefits for the media industry primarily relate to payment transactions and copyright tracking. Possible applications and technical innovations will have a far-reaching impact: content creators may be able to keep a close track of their playtimes, royalties and advertising revenues could be shared in an exact and timely manner based on consumption, and low-cost content could be purchased efficiently, even if priced at mere fractions of cents.

## 3.9. Healthcare

Using Blockchain to improve health information exchange (HIE) could unlock the true value of interoperability (the ability of computer systems or software to exchange and make use of information). Blockchain-based systems have the potential to reduce or eliminate the friction and costs of current intermediaries. Offering a promising new distributed framework to amplify and support the integration of healthcare information across a range of uses and stakeholders.

Next, table 15, it will be addressed several existing pain points and presents how Blockchain could make the system more efficient, disintermediated, and secure.

| HIE pain points | Blockchain opportunities |
|---|---|
| **Establishing a trust network** depends on the HIE as an intermediary to establish point-to-point sharing and "book-keeping" of what data was exchanged. | **Disintermediation of trust** likely would not require an HIE operator because all participants would have access to the distributed ledger to maintain a secure exchange without complex brokered trust. |
| **Cost per transaction** gave low transaction volumes reduces the business case for central systems or new edge networks for participating groups. | **Reduced transaction costs** due to disintermediation, as well as near-real-time processing, would make the system more efficient |
| **Master Patient Index** (MPI) challenges arise from the need to synchronize multiple patient identifiers between systems while securing patient privacy. | **A distributed framework for patient digital identities** which uses private and public identifiers secured through cryptography creates a singular, more secure method of protecting patient identity. |
| **Varying data** standards reduce interoperability because records are not compatible between systems. | **Shared data** enables near real-time updates across the network to all parties. |
| **Limited access to population health data** as HIE is one of the few sources of integrated records. | **Distributed, secure access** to patient longitudinal health data across the distributed ledger. |
| **Inconsistent rules and permissions** inhibit the right health organization from accessing the right patient data at the right time. | **Smart contracts** create a consistent, rule-based method for accessing patient data that can be permissioned to selected health organizations. |

*Table 15: HIE pain points vs Blockchain opportunities adapted from (Krawiec et al., 2016)*

The current state of healthcare records is disjointed due to a lack of common architectures and standards.

Information stored on the Blockchain could be universally available to a specific individual through the Blockchain private key mechanisms, enabling patients to share their information with healthcare organizations much more seamlessly. This deployment of a transaction layer on the Blockchain (figure 13) can help accomplish interoperability goals while creating a trustless, and collaborative ecosystem of information sharing to enable new insights to improve the efficiency of the nation's health care system and health of its citizens.

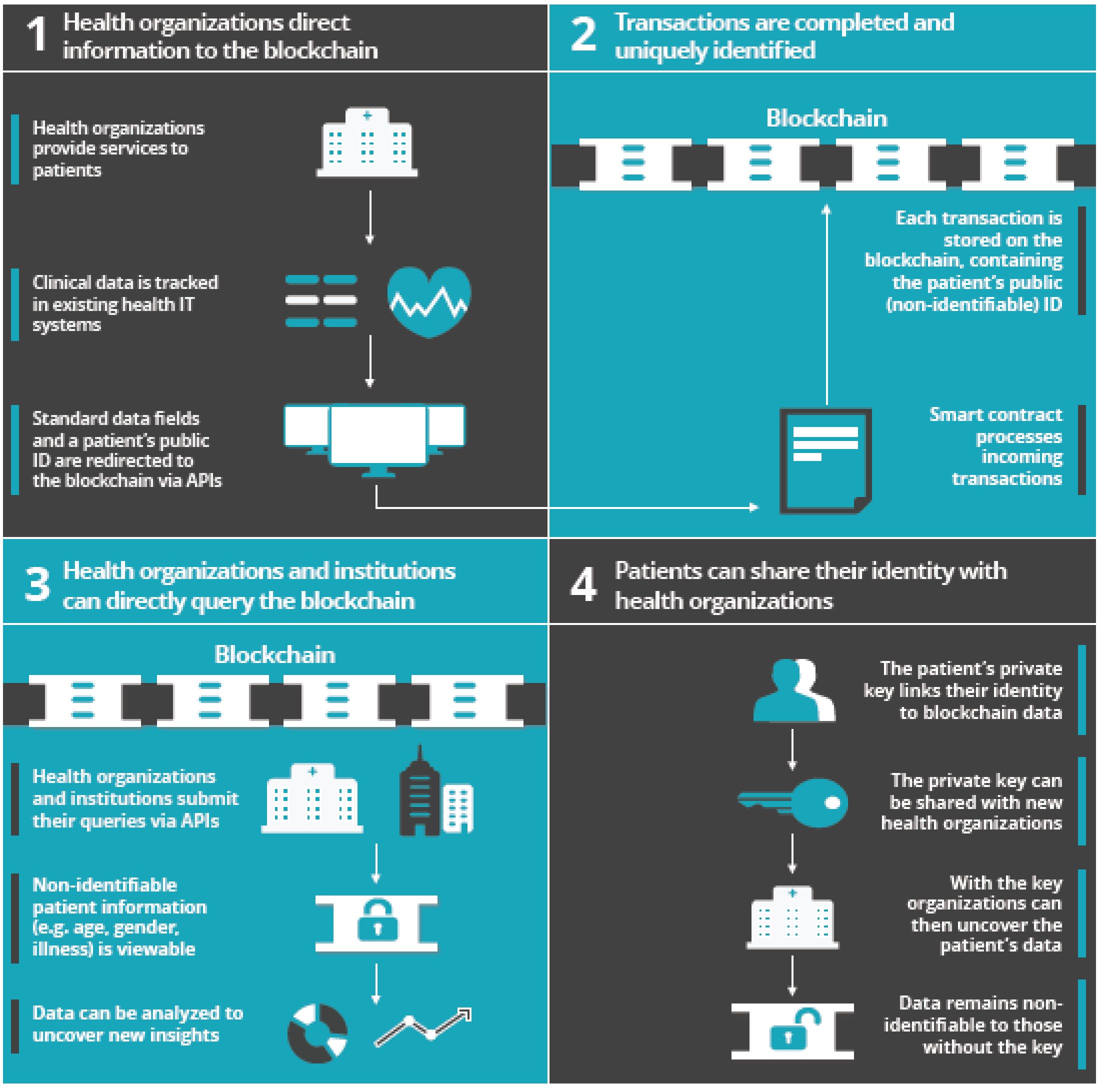

1 Health organizations direct information to the blockchain
Health organizations provide services to patients
Clinical data is tracked in existing health IT systems
Standard data fields and a patient's public ID are redirected to the blockchain via APIs
2 Transactions are completed and uniquely identified
Blockchain
Each transaction is stored on the blockchain, containing the patient's public (non-identifiable) ID
Smart contract processes incoming transactions
3 Health organizations and institutions can directly query the blockchain
Blockchain
Health organizations and institutions submit their queries via APIs
Non-identifiable patient information (e.g. age, gender, illness) is viewable
Data can be analyzed to uncover new insights
4 Patients can share their identity with health organizations
The patient's private key links their identity to blockchain data
The private key can be shared with new health organizations
With the key organizations can then uncover the patient's data
Data remains non-identifiable to those without the key

*Figure 12: Illustrative Healthcare Blockchain Ecosystem (Krawiec et al., 2016)*

### Toward Blockchain interoperability

As a transaction layer, there are two types of information that Blockchain can store, they are "On-chain" data that is directly stored on the Blockchain or "Off-chain" data with links stored on the Blockchain that act as pointers to information stored in separate, traditional databases. Storing medical information directly on the Blockchain ensures that the information is fully secured by the Blockchain's properties and is immediately viewable to those permissioned to access the chain; at the same time, storing large data files slows block processing speeds and presents potential challenges to scaling the system.

Creating interoperability requires frictionless submission and access to view data. As such, the Blockchain could serve as a transaction layer for organizations to submit and share data using one secure system

Once a standardized set of healthcare information is established, the specific data fields can be created in a smart contract to employ rules for processing and storing information on the Blockchain, as well as stipulating required approvals prior to Blockchain storage. Each time a patient interaction occurs, healthcare organizations will pass information to the smart contract.

| | **On chain data** | **Off-chain data** |
|---|---|---|
| *Data types* | Standardized data fields containing summary information in text form<br>(e.g. age, gender). | Expansive medical details and abstract data types<br>(e.g. notes, MRI images). |
| *Pros* | Data is immediately visible and ingestible to all connected organizations, making Blockchain the single source of truth. | Storage of any format and size of data. |
| *Cons* | Constrained in the type and size of data that can be stored. | Data is not immediately visible or ingestible, requiring access to each healthcare organization's source system for each record.<br>Requires Off-Chain micro-services and additional integration layers.<br>The potential for information decay on the Blockchain. |

*Table 16: On chain vs Off-chain data adapted from (Krawiec et al., 2016)*

### Blockchain strengthens data integrity and patient digital identities

In 2015, there were 112 million healthcare record data breaches due to hacking/IT incidents and In 2016, it is estimated that one in three healthcare recipients will be a victim of a data breach. An interoperable Blockchain could strengthen data integrity while better protecting patient's digital identities. (Krawiec et al., 2016).

The Blockchain's inherent properties of a cryptographic public/private key access, proof of work, and distributed data create a new level of integrity for healthcare information. Additionally, all health care organizations connected to the Blockchain can maintain their own updated copy of the health care. This feature improves security and can help limit the risk of the malicious activity, because changes are immediately broadcast to the network, and distributed ledgers provide safeguard copies against harmful hacks.

# CHAPTER 4 - BLOCKCHAIN LIMITATIONS AND CHALLENGES

The Blockchain technology is in the early stages of development, and there are still many different kinds of limitations that can prevent the technology from being globally adopted. In this chapter, it will be described examples of technical, legal and socio-economic limitations and challenges that are considered crucial to overcoming.

## 4.1. Technical Limitations and Challenges

There are several technical challenges to be overcome in order to evolve into the next phases of Blockchain mass adoption. There are seven limitations that are considered priorities to researchers (Swan, 2015). These seven limitations had been accepted as crucial challenges to be solved by general community of Blockchain developers who constantly research in order to find new solutions to these problems (Yli-Huumo, Ko, Choi, Park, & Smolander, 2016).

### Throughput

One of the Blockchain potentials issues is throughput. The number of transactions per second is significantly lower than other transaction processing centralized networks like the ones used in Banks or other similar institutions. Using Bitcoin and VISA as an example, Bitcoin process between 3 to 7 transactions per second, on the other hand, VISA process between 2000 to 10000 transactions per second. Considering this, if the Blockchain throughput does not increase it is infeasible for cryptocurrencies to replace our current monetary system.

### Latency

The amount of time required for a transaction to be confirmed is the same as to generate a new block. In Bitcoin it takes around 10 minutes. To achieve efficiency in security, it may take an hour to check if none double spending attacks were made. Again, by comparing it with central banks it takes no more than a few seconds the perform the entire process.

### Size and bandwidth

Currently, the size of the Bitcoin network is about 185 GB, if the throughput is increased to 2000 transactions per second like Visa, and maintain the size one block at 1MB, that would mean a growth of 3.9 GB/day. There are many technologies nowadays that works with a much wider amount of data, but it can be stored in central servers or compressed, In Blockchain that's not possible for security reasons. This brings us to a difficult situation where is needed an increased throughput without jeopardizing accessibility and the possibility to run a node on a regular computer.

### Security

There are a few security issues that need to be solved in order to achieve a really trusted network, the main one is 51% attacks, in which a miner can control the Blockchain and centralize the mining hash-rate of the network, giving the possibility to double-spend assets. Another issue is the cryptographic algorithms used that might be crackable and need to be strengthened.

### Wasted resources

In consensus protocols such as PoW, the consumption of energy is immensely high, the earliest estimation is that Bitcoin current estimated annual electricity consumption is 73.12 TWh, approximately the same value as Austria. PoS and other consensus protocols are being developed to solve this problem.

### Usability

The API for working with Bitcoin and other Blockchain platforms are far less user-friendly than the current standards of other easy-to-use modern APIs, such as the widely used REST APIs. It also should be created more development and testing tools and documentation to promote Blockchain fast development.

### Versioning, hard forks and multiple chains

Blockchain infrastructure should also be the object of attention by developers. Smaller chains with fewer nodes are more susceptible to 51% attacks. Another issue to be solved are chains that are split for administrative or versioning purposes, because there is no easy way to merge or cross-transact on forked chains.

## 4.2. Legal Limitations and Challenges

The technical challenges are not the only barrier to a wider adoption of Blockchain. Some of the core properties of Blockchain conflict with our legislation. Next, it will be present a list of legal limitations that need to be taken into account by governments and organizations if the technology is to be legalized and accessible to everyone (Jessica, Duncan, & John, 2017).

### Jurisdiction

Blockchain can cross jurisdictional boundaries as the nodes of the network can be located anywhere in the world. This can pose many complex jurisdictional issues which require careful consideration in relation to the relevant contractual relationships.

The principles of contract and title differ across jurisdictions and therefore identifying the appropriate governing laws is essential. However, in a decentralised environment, it may be difficult to identify the appropriate set of rules to apply.

### Liability

The risk to customers of a systemic issue with trading related infrastructure such as Blockchain could be serious if trades are not settled or are settled incorrectly. Likewise, the risk relating to security and confidentiality will be towards the top of the risk issues of any prospective customer.

Blockchain poses different risks as a consequence of the technology and manner of operations, one of the main issues affecting public Blockchain is the inability to control and stop its functioning. So, the allocation and attribution of risk and liability in relation to a malfunctioning Blockchain service must be thought through carefully, not just at the vendor-customer level, but between all relevant participants.

### Data Privacy

One of the main aspects of the Blockchain is that data is tamper-proof. Once data is stored it cannot be altered. This clearly has implications for data privacy, particularly where the relevant data is personal data or metadata sufficient to reveal someone's personal details.

Equally the unique transparency of transactions on the Blockchain is not easily compatible with the privacy needs of the banking sector: the use of crypto-addresses for identity is problematic as no bank likes providing its competitors with precise information about their transactions and the banking secrecy must be kept by law.

### Decentralised Autonomous Organisations (DAOs)

DAOs are essentially online, digital entities that operate through the implementation of pre-coded rules.

These entities often need minimal to zero input into their operation and they are used to execute smart contracts, recording activity on the Blockchain. Modern legal systems are designed to allow organisations, as well as actual people, to participate. Most legal systems do this by giving organisations some of the legal powers that real people have. But what legal status will attach to a DAO? Are they simple corporations, partnerships, legal entities, legal contracts or something else?

Since the DAOs management is conducted automatically, legal systems would have to decide who is responsible if laws are broken.

### The Enforceability of Smart Contracts

Since smart contracts are prewritten computer codes, their use may present enforceability questions if attempting to analyse them within the traditional ‘contract’ definition. This is particularly true where smart contracts are built on permissionless Blockchains, which do not allow for a central controlling authority. Since the point of such Blockchains is to decentralize authority, they might not provide for an arbitrator to solve any disputes that arise over a contract that is executed automatically.

Customers should ensure that smart contracts include a dispute resolution provision to reduce uncertainty and provide for a mechanism in the event of a dispute.

### Compliance with financial services regulation

Many sourcing arrangements, including the use of certain technology solutions, require regulated entities to include in the relevant contracts a series of provisions enabling them to exert control and seek to achieve operational continuity in relation to the services to which the contracts relate. Using Blockchain, this may well be more of a challenge. The contracts and overall arrangement will need to be carefully reviewed to ensure compliance, as required.

### Is data on a Blockchain property?

Using common law as a general principle, there is no property right in the information itself, but while individual items of information do not attract property rights, compilations of data may be protected by intellectual property rights.

When a database of personal information is transacted, if the receiver wants to use the personal information for a new purpose, in order to comply with the Data Protection Act, they will have to get consent for this from the individuals concerned.

### Due diligence on Blockchain

Public companies and private investors have already begun to make significant capital investments in Blockchain technology startups. This trend is likely to accelerate as commercial deployments of Blockchain technology become a reality. Transactional lawyers who are tasked with performing due diligence on the buy and/or sell side in connection with these investments need to understand Blockchain technology and the emerging business models based on the technology.

Traditional due diligence approaches may need to be adapted. For example, there will be unique issues concerning ownership of data residing on decentralised ledgers and intellectual property ownership of Blockchain-as-a-service offerings operating on open source Blockchain technology platforms.

## 4.3. Social-Economic Limitations and Challenges

Although Blockchain technology can provide many economics benefits (S. Ben Dhaou, Zalan, & Toufaily, 2017), it can also have a great impact on our society and economic stability. Next, it will be present some of the major social and economic impacts of Blockchain that drive researchers to find a solution (S. I. Ben Dhaou, 2018)

### Social impact

In this section is described some of the social impacts created by Blockchain that should be taken into account by researchers in order to find a solution or option to minimize those impacts.

### Absence of censorship

Since the Blockchain is public and anyone can interact with it and there is no entity regulating what can be deployed on the chain, it is not possible to prevent someone to deploy something that can be found as immoral or not acceptable such as hate speech spreading, defamation, private or confidential information, etc.

### Immutability

The Blockchain technology has the immutability properties which ensures that the data will never be changed, this increases its security but at the same time brings an issue. If a mistake was made and deployed on the Blockchain, it is impossible to correct that mistake.

### Right to be forgotten

The "Right to be forgotten" concept has been put into practice in Europe Union and other countries, which provide each citizen with the possibility to remove from the network some information regarding him. This is impossible to achieve if the information were deployed on the Blockchain.

### Lack of regulation

The fact that no governmental or other regulatory entity has control over the Blockchain, which makes possible to achieve a decentralized and distributed network, also creates a challenge, in case of an attack the victim does not have any entity to turn to compensate for the damages or to penalize the author of the attack.

### Cultural

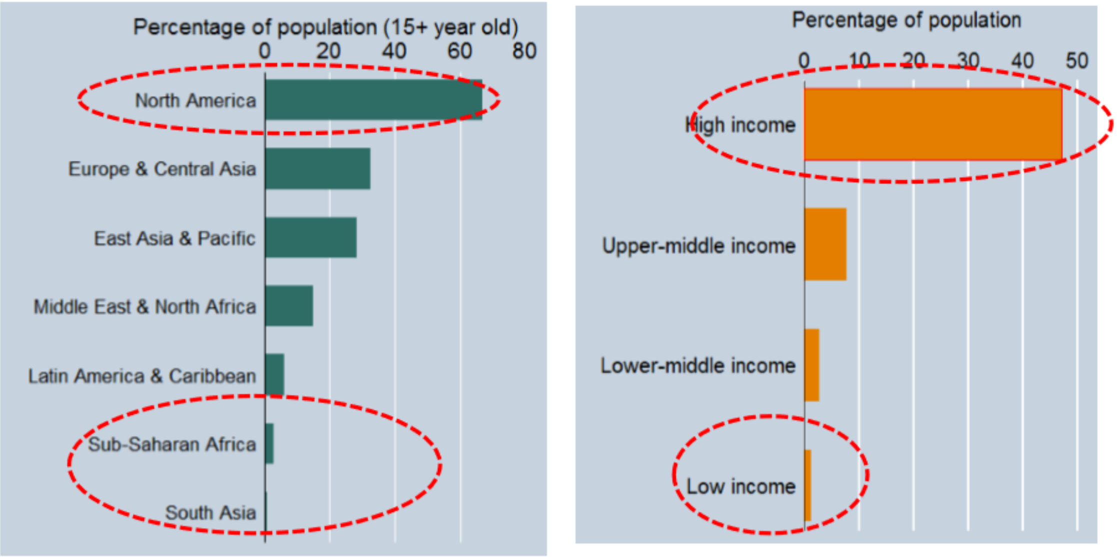
Each society has their own principles and ideals. Although in some societies it is acceptable for their personal data or properties being exposed on a transparent network, for others, it is considered an abuse of their privacy and will be not accepted.

## Economic impact

In this section is described some of the economic impacts created by Blockchain that may limit or even prevent the technology from being adopted and used.

### Access limited to a niche of users

Although Blockchain its open, it is not possible for everyone to use it and take advantage of its features. By using as an example electronic transactions, it is possible to deduct that Blockchain transaction will follow the same pattern.



**Original variable : Used electronic payments to make payments (% age 15+)**
**Variable code : [w1] [WP11633.1]**
**Source :** IMF Statistics

*Figure 13: Electronic transaction- the percentage of the population (2011-2014)* (Dhaou, 2018)

As it can be seen, there is a substantial gap when comparing geographically or by earnings-related, it is necessary to find different approaches or solutions to decrease this gap or it will be expected a greater barrier of technology acceptance

### Economic hype

While the number of investments made on Blockchain platforms increases, some of these investments are not well planned, creating more opportunities for users to invest their money on DAOS or other smart contracts that may contain code errors, resulting on the loss of their savings.

### Not adapted for the general-purpose

The Blockchain technology is not yet adapted to be used as a general-purpose currency and replace the current currency system, being unpractical to provide financial services that do not depend on the container as a general-purpose currency.

### Effectiveness, efficiency and viability not proved

Studies showed that effectiveness, efficiency and viability of the public ledger platform are not proven yet and not supported empirically.

### Volatility

The general vulnerability of the system, the risk of the currency losing value or the discovery of a new vulnerability being found and exploited by malicious users, creates great uncertainty among users to know if they should invest or not.

### Slow proof-of-work process

The throughput of Blockchain transaction is significantly lower than our current monetary systems, making very difficult for cryptocurrencies to satisfy our increasing need for nearly real-time transactions.

### The absence of a regulatory

Since the Blockchain is decentralized, in face of a problem or dispute there is no regulatory entity that may decide how to solve the dispute. This is a major challenge in order to gain the trust of users, regulating measures should be able to provide solutions to these problems

# CHAPTER 5 - BLOCKCHAIN SOLUTION EXAMPLE

This chapter has the purpose of putting into practice all the knowledge gathered in the previous chapter by developing a proof-of-concept. The chapter is composed of two sections, the first is a presentation of the chosen use case, being the second one the description of the developed decentralized application (DApp). The goal is to create a Blockchain-based voting system platform that preserves voter privacy and security using smart contracts in order to achieve voter administration and auditable voting record.

## 5.1. Electronic Voting

Voting is a crucial and serious activity for any country or organization that needs to elect a certain entity for a certain role. The most common way to vote is through a paper-based system, but it is not the safest or economical method.

The first-ever electronic voting system was introduced in the early eighties by David Shaum, Since then the method of voting has evolved over time to improve its speed, security, flexibility, availability and cost during its three main steps, being them the authentication of the voter, the casting of the vote and for last, expose the results.

“There have been several studies on using computer technologies to improve elections, These studies caution against the risks of moving too quickly to adopt electronic voting machines because of the software engineering challenges, insider threats, network vulnerabilities, and the challenges of auditing” (Kohno & Stubblefield, 2004).

### 5.1.1. Requirements of E-voting Systems

To judge the adequacy and security parameters of a voting system it is necessary to evaluate twelve core requirements (Palas Nogueira & De Sá-Soares, 2012). These requirements are presented in order, from the most important to the least (table 17).

| *Requirement* | *Description* |
|---|---|
| *Authenticity* | Only persons with the right to vote should be able to cast a vote |
| *Singularity* | Each voter should be able to vote only once |
| *Anonymity* | It should not be possible to associate a vote to a voter |
| *Integrity* | Votes should not be able to be modified or destroyed |
| *Uncoercibity* | No voter should be able to prove the vote that has cast |
| *Verifiability* | Anyone should be able to independently verify that all votes have been correctly counted |
| *Auditability and Certifiability* | Voting systems should be able to be tested, audited and certifiable by independent agents |
| *Mobility* | Voting systems should not restrict the voting place |
| *Transparency* | Voting systems should be clear and transmit accuracy, precision, and security to voter |
| *Availability* | Voting systems should be always available during the voting period |
| *Accessibility and Convenience* | Voting systems should be accessible by people with special needs and without requiring specific equipment or abilities |
| *Detectability and Recoverability* | Voting systems should detect errors, faults and attacks and recover voting information to the point of failure |

*Table 17: Core requirements of a voting system (Palas Nogueira & De Sá-Soares, 2012)*

From the order of importance shown above, it is possible to observe the requirements with the most impact are related to the security and anonymity of the voting system and the requirements with less impact are related with the usability and transparency of its infrastructure.

### 5.1.2. Trust Factors of E-voting Systems

Trust is a crucial concept in a voting system, if all stakeholders involved in the election do not trust the voting system that they are using, then the result provided by the system probably will not be accepted as valid.

There are many cases where e-voting systems have failed, providing incorrect results based on the number of votes and electors. Those events decrease the trust of the population on e-voting systems as a substitute for the paper ballot (Palas Nogueira & De Sá-Soares, 2012).

The factors which have more weight on the public confidence in e-voting systems can be presented using a table (table 18) divided into two columns, the first is the description of the trust factor and the second one is the percentage of answers who found this factor relevant (Palas Nogueira & De Sá-Soares, 2012).

| **Trust Factor** | **Percentage %** |
|---|---|
| Different types of elections require different security levels | 33.6 |
| Audits to the system and certifications awarded to the system | 18.6 |
| Reputation and competency of the system development team | 11.5 |
| Various uses of the system without errors | 10.6 |
| The monitoring committee of the electoral process | 9.7 |
| Information and explanations about the system | 7.1 |
| Tests made to the system | 4.4 |
| Guarantee of anonymity | 2.7 |
| Transparency of the system | 0.9 |
| Open source code | 0.9 |

*Table 18: Main Factors Influencing Trust in EVS adapted from (Palas Nogueira & De Sá-Soares, 2012)*

From table 18, it is possible to assume that the better way to provide trust on e-voting systems is to address our attention to the different security levels required by different types of election, and the necessity of an efficient and effective way to audit those systems and award them with certifications based on their results.

### 5.1.3. Advantages of Blockchain on E-voting Systems

Blockchain has the potential the improve electronic voting systems by solving some of their major limitations and issues. Next, each e-voting system requirement will be addressed in order to confirm if a Blockchain solution can fulfil each one of them.

Authenticity: Every user of the network is identified by a public key, that can only be accessed by its own private key. Assuming that every voter will keep its own private key secret, then the authenticity requirement is fulfilled.

Singularity: Every vote cast is linked to the public key of the voter on the Blockchain, by allowing each public key to cast only one vote, it can assure the singularity requirement.

Anonymity: since every user is identified by a public key, and the vote stored is encrypted, is impossible to associate a voter with a vote

Integrity: since the hash pointer provides the Blockchain with tamper-evident properties, every vote stored has the same properties, meaning that it can’t be adulterated.

Uncoercibity: Every vote cast will be encrypted before being stored on the Blockchain, making it impossible for anyone to know the content of that vote.

Verifiability: since the Blockchain is transparent to every node of the network, everyone could confirm that the number of votes cast and counted are the same.

Auditability and Certifiability: equally to the verifiability requirement, the transparent property of Blockchain allow for any node of the network to audit the Blockchain. Since the system code is open source and visible on the Blockchain, means that the used application could be also audited.

Mobility: the only requirement to access the network is a device with internet connection and an address in the Blockchain platform, meaning that it is not required any kind of infrastructures or voting machines.

Transparency: identically to the Auditability and Certifiability requirement, transparency is one of the Blockchain properties, and every application implemented in the Blockchain inherits the same property.

Availability: since Blockchain is a distributed network, as long as the network have the required amount of nodes to achieve consensus, the voting system will be always available.

Accessibility and Convenience: equal to the mobility requirement, Blockchain does not require any kind of infrastructure or voting machine. Only a device with an internet connection is required.

Detectability and Recoverability: If any malicious action is made on the Blockchain, it will be detected by the network and considered invalid, this property fulfils the detectability requirement. In the case of recoverability, as soon as some data is stored on the Blockchain, no longer can be deleted, meaning that every information can always be recovered.

As can be seen, a Blockchain-based voting application can satisfy every requirement and has the potential to improve our current voting systems.

## 5.2. Proposed Solution

This section will address the solution developed, describing the chosen Blockchain platform and the components of the decentralized application developed.

Regarding the DApp, it will present the architecture of the system, the smart contracts that were deployed on the Blockchain, the interface and API and functionalities that are there implemented, and lastly, it will explain the encryption server and how it can assure the privacy of the vote. The dependencies and tools required for the development of the project are also exposed.

### 5.2.1. Blockchain Platform

The chosen Blockchain platform is Ethereum because of its alternative protocol suitable for developing decentralized applications, by building a Blockchain with a built-in turing-complete programming language, allowing users to write smart contracts with their own rules for transactions formats, rules for ownership and state transitions functions.

Ethereum is a public and permissionless Blockchain which philosophy is based on principles that are considered ideal to electronic voting systems.

#### Simplicity

The Ethereum protocol is designed to be as simple as possible, making it easier for an average programmer to follow and implement the entire specification. Any optimization that increases the protocol complexity should not be included unless it provides very substantial benefit. This principle can cost some data storage or time inefficiency.

#### Universality

Instead of having features, Ethereum provides an internal turing-complete programming language that allows the creation of any smart contract or transaction type that can be mathematically defined. Increasing the number of use cases that can be exploited by programmers.

#### Modularity

The Ethereum protocol is designed as modular and separable as conceivable, making it possible to make small protocol modifications on a specific module and keep the rest of the application intact.

#### Agility

Although Ethereum his very judicious about making modifications, their protocol is not definitive and is open to change. If an opportunity is found, it will be tested and exploited in order to achieve improvement.

## Non-discrimination and non-censorship

All mechanism created to regulate the protocol is designed to prevent harm and not to attempt to oppose a specific undesirable application. Every application can be migrated to the network as long as the programmer is willing to keep paying a per-computational-step transaction fee.

## Ethereum Mechanism

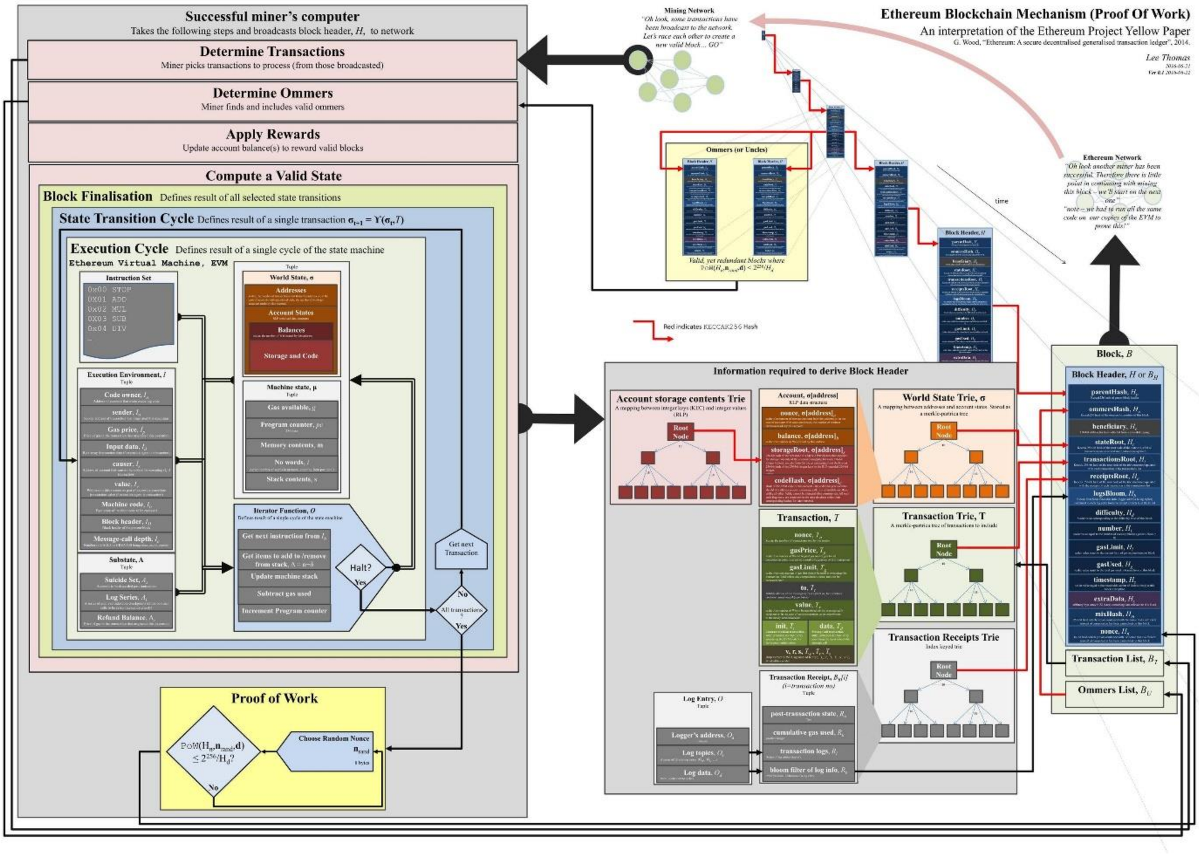

Successful miner's computer
Takes the following steps and broadcasts block header, H, to network
Determine Transactions
Miner picks transactions to process (from those broadcasted)
Determine Ommers
Miner finds and includes valid ommers
Apply Rewards
Update account balance(s) to reward valid blocks
Compute a Valid State
Block Finalisation Defines result of all selected state transitions
State Transition Cycle Defines result of a single transaction σt+1 = Υ(σt,T)
Execution Cycle Defines result of a single cycle of the state machine
Ethereum Virtual Machine, EVM
Instruction Set
0x00 STOP
0X01 ADD
0x02 MUL
0X03 SUB
0x04 DIV
Tuple
World State, σ
Addresses
Account States
Balances
Storage and Code
Machine state, μ
Tuple
Gas available, g
Program counter, pc
Memory contents, m
No words, i
Stack contents, s
Execution Environment, I
Tuple
Code owner, Ia
sender, Is
Gas price, Ip
Input data, Id
causer, Ie
value, Iv
Machine code, Ib
Block header, IH
Message-call depth, Ie
Substate, A
Tuple
Suicide Set, As
Log Series, Al
Refund Balance, Ar
Iterator Function, O
Defines result of a single cycle of the state machine
Get next instruction from Ib
Get items to add to /remove from stack
Update machine stack
Subtract gas used
Increment Program counter
Halt?
Yes
Get next Transaction
No
All transactions ?
Yes
Proof of Work
PoW(Hn,nrand,d) ≤ 2256/Hd?
No
Choose Random Nonce
nrand
8 bytes
Mining Network
"Oh look, some transactions have been broadcast to the network. Let's race each other to create a new valid block… GO"
Ethereum Blockchain Mechanism (Proof Of Work)
An interpretation of the Ethereum Project Yellow Paper
G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014.
Lee Thomas
2016-06-21
Ver 0.1 2016-06-22
Ommers (or Uncles)
Block Header, U
Valid, yet redundant blocks where PoW(Hn,nrand,d) < 2256/Hd
time
Ethereum Network
"Oh look another miner has been successful. Therefore there is little point in continuing with mining this block – we'll start on the next one"
"note – we had to run all the same code on our copies of the EVM to prove this!"
Block Header, H
Red indicates KECCAK256 Hash
Information required to derive Block Header
Account storage contents Trie
A mapping between integer keys (KEC) and integer values (RLP)
Root Node
Account, σ[address]
RLP data structure
nonce, σ[address]n
balance, σ[address]b
storageRoot, σ[address]s
codeHash, σ[address]c
World State Trie, σ
A mapping between addresses and account states. Stored as a merkle-patricia tree
Root Node
Transaction, T
nonce, Tn
gasPrice, Tp
gasLimit, Tg
to, Tt
value, Tv
init, Ti
data, Td
v, r, s, Tw, Tr, Ts
Transaction Trie, T
A merkle-patricia tree of transactions to include
Root Node
Transaction Receipts Trie
Index keyed trie
Root Node
Log Entry, O
Tuple
Logger's address, Oa
Log topics, Ot
Log data, Od
Transaction Receipt, BR[i]
(i=transaction no)
Tuple
post-transaction state, Rσ
cumulative gas used, Ru
transaction logs, Rl
bloom filter of log info, Rb
Block, B
Block Header, H or BH
parentHash, Hp
ommersHash, Ho
beneficiary, Hc
stateRoot, Hr
transactionsRoot, Ht
receiptsRoot, He
logsBloom, Hb
difficulty, Hd
number, Hi
gasLimit, Hl
gasUsed, Hg
timestamp, Hs
extraData, Hx
mixHash, Hm
nonce, Hn
Transaction List, BT
Ommers List, BU

*Figure 14: Ethereum Blockchain Mechanism https://i.stack.imgur.com/afWDt.jpg*

The previous image (figure 26) tries to represent the Ethereum mechanism, in order to facilitate the understanding of how this platform works, it will be explain its core components.

### Ethereum Accounts

On the Ethereum platform the state is made of accounts, each one has an address and contains four data fields, being them the nonce, the ether balance, the smart contract code if it exists and the account storage that is empty by default. There are two types of Ethereum accounts, being external owned accounts controlled by private keys, and contract accounts controlled by their contract code.

### Transactions

A transaction is a signed data package sent from an externally owned account. Each transaction contains the recipient of the message, the sender signature, the amount of ether to be sent, an optional data field, the startgas value and the gasprice value.

Startgas represents the maximum number of computational steps that the transaction execution is allowed to take. Gasprice represents the fee paid per computational step. These two fields are crucial in order to prevent infinite loops or other computational wastage in code.

Gas is the fundamental unit of computation, every operation that “writes” on the Blockchain consumes gas, and each transaction has a 5 gas cost for every data byte. On the other hand, “reads” from the Blockchain doesn't have any cost. The reason for this is to make an attacker pay proportionately for every recourse that consumes.

### Messages

A message is like a transaction, but instead of being produced by an external account it is produced when a contract currently executing code produces and executes a message. Each message contains the sender address, the recipient of the message, the amount of ether to be sent, an optional data field and a startgas value.

### Ethereum State Transaction

State transactions are direct transfers of value and information between accounts (figure 16). The function to update the state can be described in the following way, checks if the transaction Is well-formed, if so, it calculates the transaction fee by multiplying the stratgas by the gasprice and add a certain quantity of gas per byte to pay for the bytes in the transaction. Assuming the sender can pay the expected amount, it will be removed from his balance. Then the transaction will be transferred from the sender account to the receiver. If any of this passes failed the state will be reverted, except for the payment fees that will still be sent to the miner.

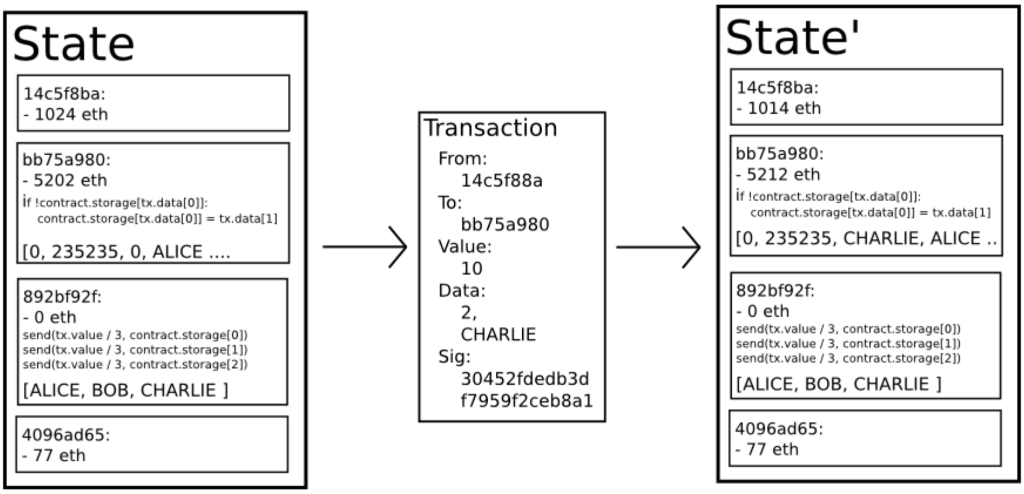

State
14c5f8ba:
- 1024 eth
bb75a980:
- 5202 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, 0, ALICE ....
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE ]
4096ad65:
- 77 eth
Transaction
From:
14c5f88a
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1
State'
14c5f8ba:
- 1014 eth
bb75a980:
- 5212 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, CHARLIE, ALICE ..
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE ]
4096ad65:
- 77 eth

*Figure 15: Ethereum State Transition Function representation* (Ethereum, n.d.)

### Code Execution

Ethereum contracts are written in a low-level, stack-based bytecode language called Ethereum virtual machine code, it consists in a series of bytes where each byte represents an operation that is executed in an infinite loop until an error is returned,

The operations have access to three types of space to store data such as stack, a container that follows "last-in-first-out" logic, a memory that is an infinitely expandable byte array, and for last a storage that unlike memory and stack, doesn't reset after computation but persists for the long-term instead.

### Blockchain and Mining

Ethereum and Bitcoin are very similar, but on Ethereum, blocks not only contain a copy of the transaction list, like Bitcoin but also hold a copy of the most recent state (figure 17).

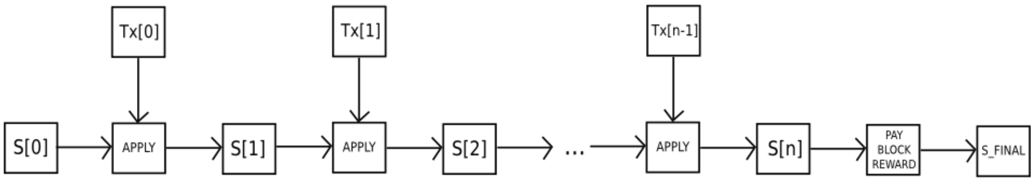


*Figure 16: Ethereum Blockchain representation* (Ethereum, n.d.)

In order to validate a block, the algorithm follows the next steps. First, some conditions have to be confirmed, such as the existence and validation of the previous block referenced. The time stamp of the block generated has to be greater than the previous and less than 15 minutes into the future. The block number, difficulty, transaction root, uncle root and gas limit also need to be valid, and for last, confirm that the proof-of-work on the block is valid too.

After all these conditions been verified it will proceed to validate the block. Let's assume that S[0] is the state at the end of previous block and Tx will be the block transaction list with n transactions. As it can be see from figure 28, if none of the applications returns error or the gas limit is exceeded, then for all i in 0…n-1, S[ i+1 ] = APPLY(S[i], Tx[i]).

The final state (S_FINAL) is equal to S[n] but the block reward was already paid to the miner. After all the previous steps checked then it is required to confirm if the Merkle tree root of the S_FINAL is equal to the final state root provided in the block header, if so, the block is validated.

### 5.2.2. Decentralized Application

The research on already developed or in development stage voting dapps (Gregory; Dapp University, n.d.), (Karl Floersch, 2016), (McCorry, Shahandashti, & Hao, 2017) and the most relevant one (Dagher, Marella, Milojkovic, & Mohler, 2018), and taking into account the requirements of an electronic voting system, led to consider this the best solution possible for this problem.

### Architecture Design

The design used to develop this decentralized application is multi-contract & multi-state, where different contracts have different functionalities. They are still dependent on each other for functionality and it is possible for some contracts to exist longer than others (ConsenSys, n.d.).

This approach has the advantages of modularity. Since the functionalities are separated in modules it is not required to modify the whole project in order to change some functionalities. The consequence for this design is the tendency to be more computationally expensive due to modularity.

### Application architecture

This implementation consists of an HTML interface for the application users, a cryptographic server that will encrypt/decrypt the votes made, three contracts deployed on the Ethereum Blockchain that are coded in Solidity language and an API to act as a bridge between all the components mentioned before. The architecture of the system is represented in figure 18.

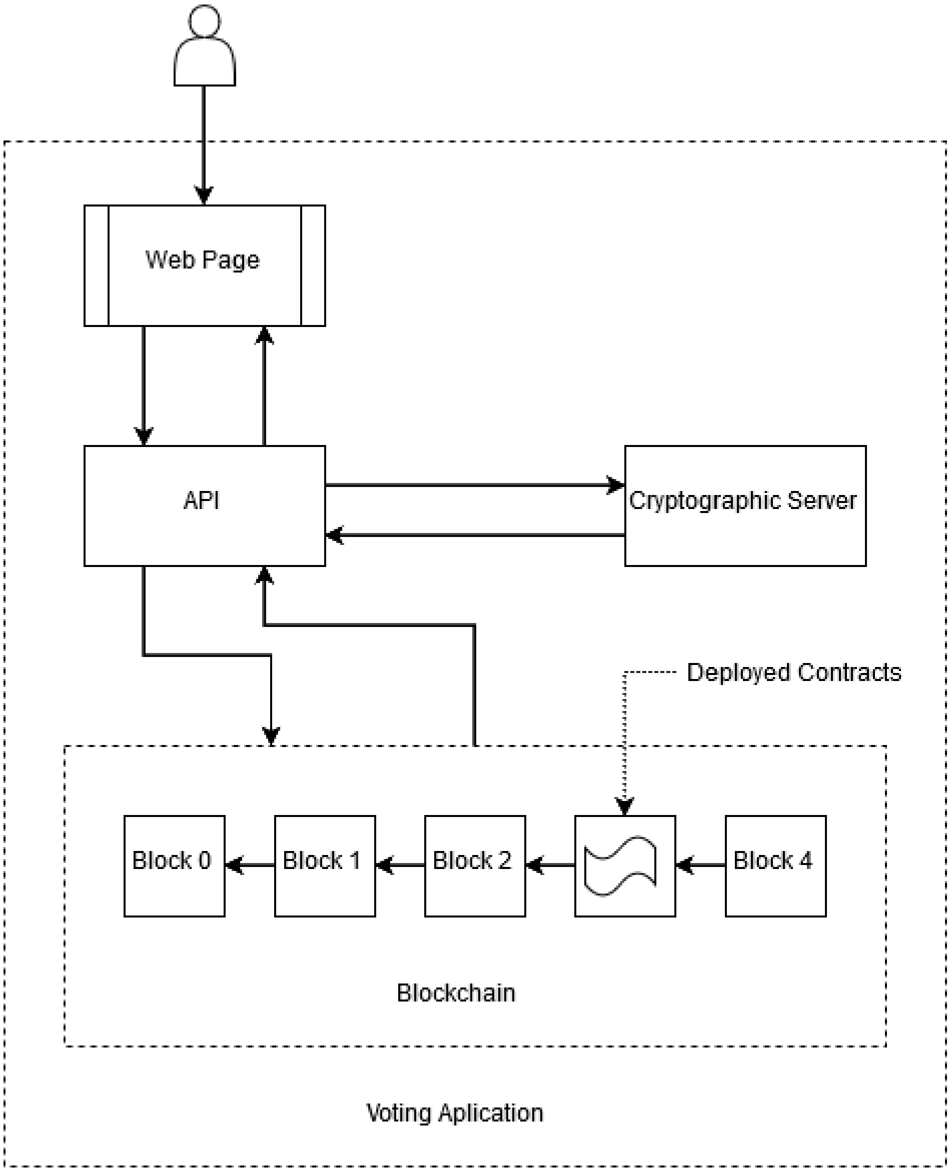


*Figure 17: Representation of the system architecture*

### 5.2.3. Users

There are three types of users who can interact with the application, being them the administrator, the creator and the voter. The administrator has the responsibility for the initial deployment of the contracts and grant or revokes the permission for a user to create ballots. The creator is a user who was granted permission for creating new ballots. The voter, as the name indicates, is the user who can vote for a candidate in a certain ballot.

### 5.2.4. Contracts

In order to achieve a better performance, the application functionalities are divided into three contracts, being them Record, Creator and Election. Each one of them with a specific purpose but they all work together to achieve the same goal. On figure 19, it is possible to analyse the memory field structure of each smart contract, the lines found between fields represent relational data.

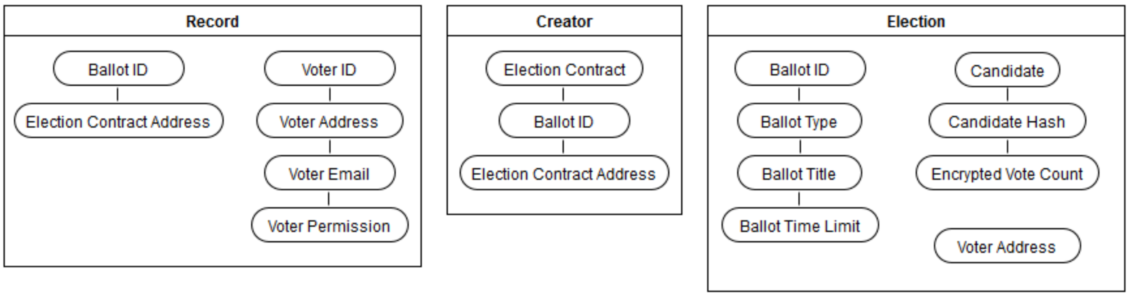


*Figure 18: Representation of the deployed smart contracts*

#### Record

This contract will store information related to every voter and creator registered and ballots created, being the owner of this contract the administrator.

When deploying this contract, the administrator needs to specify the users email domains that has permission to register, posteriorly, it can add new domains if required. Every user who are able to register on the system will have his information's stored on this contract, this is necessary for validations functions. When registering the user can request for creating permissions, if granted by the administrator, the user will be able to create new ballots, this mean a new Election contracts. For every ballot created, its ID and Ethereum address will be linked and stored.

The code of this contract can be found on appendix A.

#### Creator

This contract will spawn new Election contracts when required by a user with permissions to do it, the parameters of the new ballot need to be delivered by the creator. The owner of this contract is the administrator. The code of this contract can be found on appendix B.

#### Election

This contract will be our election, storing its parameters, candidates and votes. The owner of this contract is its creator.

When deploying this contract, the creator needs to specify all the parameters regarding the election contract, being them the ID, the type, the deadline and the title of the ballot. After introducing and validating the candidates, assuming that the deadline was not exceeded, it is possible to vote. First, the voter information will be verified, if valid, the vote will be encrypted, and the current vote count updated. When finished the deadline, in case of an election type, the vote count will be decrypt and the result revealed. While on a poll type, the current result is always available.

The code of this contract can be found on appendix C.

### 5.2.5. Interface and API

#### Interface

The user interfaces is an HTML page that will allow users to access the functionalities of the application and insert the required information to invoke them. The code of this contract can be found on appendix D.

#### API

The API is responsible to react to actions made on the interface and interact with the encryption server and the Blockchain. The code of this contract can be found on appendix E.

For each request made in the interface, it will interact with the encryption server by server calls to encrypt, decrypt or add votes. to interact with the Blockchain it is required transactions or web3.eth.calls, in order to store or retrieve information, respectively.

The web3.eth.call executes a message call transaction, which is directly executed in the virtual machine of the node, but is never mined into the Blockchain, by doing so it is possible to retrieve information from the Blockchain without paying startgaz.

### Functionalities

Next, it will be present the functionalities that are available to users and how the system will process them when they are invoked. In figure 20, a use case diagram is presented to demonstrate which users can access each one of these functionalities.

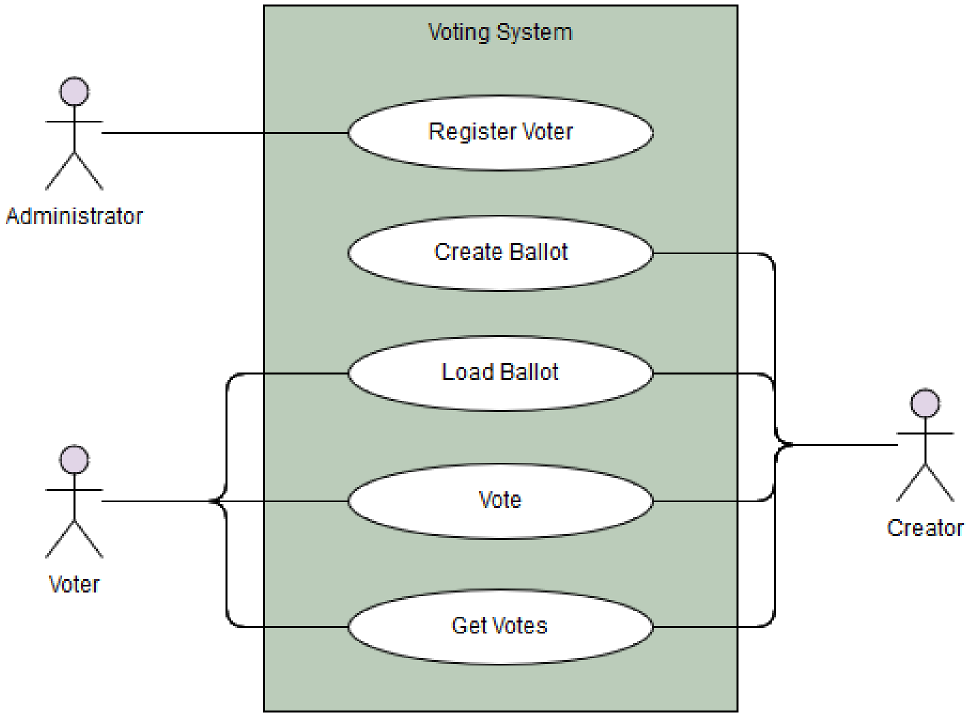


*Figure 19: Use Case Diagram*

### Register Voter

For a user to be able to register it is required for him to insert on the interface his email, institutional number and inform if he requests a creator permission. The API will receive this information from the interface and will send an eth.call to the Record contract to verify the user information by checking if the domain of the email is equal to the one specified when the contract was deployed and if the user was not already registered. If all verification are validated, the user will receive a notification on the interface and the API will send a transaction to the contract to register the new voter and store his information.

### Create Ballot

In order to create a new ballot a user, with permissions to do it, have to insert his email and the parameters of the new ballot on the interface. It is necessary to specify the new ballot title, the deadline date and time, the candidates and if it will be a poll or an election. In the first option the number of votes of each candidate is always available, while on the second option the results will be available after the deadline imposed by the creator.

After submitting this information, the interface will send it to the API that will proceed by sending two eth.call to the Record contract to verify if the creator email and Ethereum address are valid, if so, the API will send a third eth.call to verify if the user has permissions to create ballots.

After confirming that all verifications are valid, the API will send a transaction to the Creator contract with a request to create a new Election contract with the parameters previously inserted by the creator along with a new 32-bit random generated ballot ID. Once the new Election contract has been deployed it will return its address to the Creator contract.

Next, the API will send another eth.call to the Creator contract to retrieve the address of the Election contract that has been deployed and will send a transaction to the Record contract to store that address along with the corresponding ballot ID. After this process is completed it will be displayed on the interface the ballot ID, that must be written down by the creator and shared with the users who want to vote on that ballot.

### Load Ballot

By inserting the ballot ID provided by the Election contract creator on the interface, the API will send an eth.call to the Record contract to verify if the ID is valid, if so, it will load the respective ballot and show the title, the name of the candidates and the respective votes if the ballot type was poll, if not, the votes will remain blank until the end of the deadline. In order to present the number of votes of each candidate, the API needs to send a request to the Encryption server to decrypt the vote count.

### Vote

After loading the ballot, the user can vote by providing his email and choosing one of the available candidates. The API will receive this information from the interface and send an eth.call to the Record contract to verify if the voter is valid, if the voter passes the verification it will send an eth.call to the Election contract to verify if the chosen candidate is valid and if the vote was done before the ballot deadline. If the current block timestamp is greater than the deadline the vote will be not considered valid.

If all verifications are validated the vote will be stored as an array being 1 for the chosen candidate and 0 for the remaining options, the vote will be sent to the Encryption server to be encrypted. As soon as the vote has been encrypted, the previously encrypted vote count will be retrieved from the Election contract by the API using an eth.call and sent to the Encryption server to be homomorphically added together, resulting in the new encrypted vote count.

#### Get Votes

Every time a voter loads the ballot or make a vote the API will send an eth.call with the hashed candidates to retrieve the current encrypted vote count. Depending on the deadline and if it is a poll or election, it would either decrypt the votes and display them or it will wait for the end of the deadline.

### 5.2.6. Encryption Server

The solution to ensuring the privacy of the votes requires to prevent the unauthorized access to the vote. To achieve that, each vote is encrypted. The code of this contract can be found on appendix F.

Encryption algorithms are divided into two categories, being the private key and public key. On private key encryption schemes, the message is encrypted and decrypted using the same key, while on public key encryption schemes the message is encrypted by the public key and decrypted by the private key (Yi, Paulet, & Bertino, 2014).

#### Homomorphic Encryption

The definition of homomorphism is a structure-preserving map between two algebraic structures, such as groups. Homomorphic encryption is a public key encryption scheme with properties that allow specific types of computations to be carried out on ciphertext, generating an encrypted result, when decrypted the result will be equal to perform the same operation on the plaintext. Making possible to operate on messages without ever releasing their content (Yi et al., 2014).

Using as an example (P, C, K, E, D) as an encryption scheme, being P the plaintext and C the ciphertext, the K will be the key and E, D the encryption and decryption algorithms. Assuming that the plaintexts form a group and the ciphertexts form a group then the encryption algorithm E is a map from the group P to the group C. For all a and b in P and k in K, the encryption scheme is homomorphic if fulfil the following function.

$$E_k(a) \circ E_k(b) = E_k(a \diamond b)$$

### Paillier Cryptosystem

The Paillier Cryptosystem is a modular, public key encryption scheme, created by Pascal Paillier in 1999. This cryptosystem is based on the belief that the decisional composite residuosity (DCR) assumption is intractable (Yi et al., 2014).

The DCR problem states that, given composite N and an integer z, it is hard to decide whether z in an N-residue module N$^2$ or not, so if there exits y such that

$$z = y^n (mod\, n^2)$$

The Paillier encryption scheme is composed of three functions, being the key generation, the encryption and decryption algorithms (Yi et al., 2014).

Key Generation

In order to generate a key, it is required to randomly choose two large prime numbers p and q.

$$n = pq, \qquad \lambda = lcm(p-1, q-1)$$

Lcm stands for a least common multiple that means the smallest positive integer that is divisible by both (p-1) and (q-1).

Next, it is required to randomly select an integer g where $g \in \mathbb{Z}_{n^2}^*$.

$$\mu = (L\left(g^{\lambda}(mod\ n^2)\right))^{-1}(mod\ n)$$

The function L is defined as

$$L(u) = \frac{u-1}{n}$$

Completed the previous equations, the result will be the public key required for encryption (n, g) and the private key required for decryption $(\lambda, \mu)$.

### Encryption

Assuming that m is the message to be encrypted, referred to as plaintext, where $m \in \mathbb{Z}_n$.

Next, it requires to randomly select r where $r \in \ \mathbb{Z}_n^*$.

Assuming c is the generated ciphertext, the function to encrypt the plaintext is

$$c = g^m \cdot r^n (mod\ n^2)$$

### Decryption

In order to decrypt our ciphertext and return the original message, it is used the following function

$$m = L\left(c^{\lambda}(mod\ n^2)\right) \cdot \ \mu(mod\ n)$$

It is important to notice that decryption is essentially one exponentiation module $n^2$ (Yi et al., 2014).

The Paillier scheme inherits the same properties as the Homomorphic encryption, such as the homomorphic addition and multiplication of plaintexts. These properties make the Paillier encryption scheme very malleable and provide semantic security against chosen-plaintext attacks, however, its malleability makes it unprotected against adaptive chosen-ciphertext attacks. So, it is necessary to have extra attention when deciding where to use this encryption scheme. In an application such as secure electronic voting and threshold cryptosystems, these properties may indeed be necessary (Yi et al., 2014).

### 5.2.7. Dependencies and Tools

In order to develop, run and test this project, there are some dependencies and tools required, such as:

Node.js is an open source server environment, as an asynchronous event-driven JavaScript runtime. It runs single-threaded, non-blocking, asynchronously programming, which is memory efficient.

NPM is a package manager for Node.js, where each package contains all the files you need for a module. A module is a JavaScript library.

Lite server is a lightweight node server used for the development of the web app. It can open the app on the browser, refreshes when HTML or javascript files are changed, injects CSS changes using sockets, and has a fallback page when a route is not found.

Truffle is a development framework for Ethereum, making it easier to manage smart contract lifecycles, to write automated tests for contracts, to write simple, manageable deployment scripts, and a simpler network management. Truffle also provides a powerful interactive console that includes access to all built contracts and all available Truffle commands.

Ganache is a personal Blockchain for Ethereum development that can be used for deploying contracts, develop applications and run tests. By using Ganache is possible to see the current status of all accounts, the log output of Ganache's internal Blockchain and examine all blocks and transactions. Ganache is the previous TestRPC.

Metamask is an add-on that can be installed on most browsers, acting as a bridge between the browser and the Ethereum Blockchain. By using Metamask is possible to run Ethereum DApps without running a full Ethereum node. It also includes a secure identity vault, providing a user interface to manage your identities on different sites and sign Blockchain transactions.

Remix IDE is a powerful, open source tool that can be used on the browser or locally. Its purpose is to help writing smart contracts coded in Solidity. It also supports the deployment of those contracts, tests and has a very useful debugging tool.

### Accessing the application code

The developed aplication project can be found on the following github repositorium along with a "read me" file explaining how to run it on your local host machine.

https://github.com/Jorge-Lopes/Blockchain_E-voting_Dissertation

# CHAPTER 6 - CONCLUSION

In conclusion, Blockchain is a versatile technology that can be implemented in a vast list of industries and maybe change completely the way it approach some of their use cases. Recapping the information gathered during this dissertation, it was revealed the main concepts and particularities of the Blockchain technology, how can it be applied to the industry and the limitations associated.

Blockchain can be described as a distributed data structure that is replicated and shared among the members of a network, whose purpose is to record every transaction done. Each transaction is batched into timestamped blocks and each block is identified by its cryptographic hash. On each block the hash of the previous one is stored, creating a link between the blocks, or as the name implies, a chain of blocks, creating this way, a transparent and immutable history of records whose veracity is provided by a consensus protocol.

The link that connects each block is a cryptographic hash function that stores the address and the encrypted header of the previous block, by doing so, it can provide the Blockchain with tamper-evident properties. The consensus protocol is the mechanism that allows a decentralized network to arrive at an agreement about the state of the Blockchain and forces all nodes to behave accordingly to the network principles.

Smart contracts are an autonomous agent stored in the Blockchain that allows us to have general purpose computation occur on the chain, their main purpose is to manage data-driven interactions between entities on the network. By coding the necessary rules and interactions is possible to apply Blockchain to a vast group of industries and organizations. The main focus of this technology has been financial services, although, there are many others sectors that could be improved such as retail and consumer goods, energy and power, manufacturing, governmental organizations, healthcare, and many others. For all referred industries, have been described in this document, how can Blockchain improve some of their use cases by eliminating the need for third entities to regulate the network, making each process more efficient and economical.

Undoubtedly, Blockchain technology unique properties have an extremely high potential to transform the future, however, just like every new technology, it has a number of disadvantages that should be mentioned and studied in order to find a solution to overcome them. Those limitations can be divided into three categories, being them technical, legal or social-economic. The technical limitations are the ones more directly connected to the performance and success of the technology, such as the throughput, size and bandwidth of the Blockchain. Legal limitations focus on some properties of Blockchain that conflict with our legislation, such as liability and data privacy. Lastly, the social-economic limitations that can also have a great impact on our society and economic stability, an example of those limitations being the absence of censorship, immutability, economic hype and volatility.

To synthesize all the knowledge gathered, a proof of concept was developed. The goal was to create a Blockchain-based voting system that preserves voter privacy and security to create a voting framework that utilizes a Blockchain platform and smart contracts in order to achieve voter administration and auditable voting record.

After deciding the main requirements and trust factors for an electronic system, a decentralized application was developed considering_those same requirements. The chosen Blockchain platform was Ethereum because of its alternative protocol adapted for developing decentralized applications, by building a Blockchain with a built-in turing-complete programming language (Solidity), allowing users write smart contracts where they can set their own rules for transactions formats, rules for ownership and state transitions functions. Ethereum is a public and permissionless Blockchain which philosophy is based on principles that were considered ideal to electronic voting systems.

The architecture of the developed DApp is described along with the necessary contracts and other core components required to achieve a good performance. To ensure that the application provides vote privacy it was necessary the use of an encryption server with homomorphic properties, more specifically, a Paillier encryption scheme. The dependencies and tools required for the development of this application were also presented in this document.

It was possible to conclude that Blockchain has the potential the improve electronic voting systems by solving some of their major limitations and issues. By adopting the Blockchain is possible to enhance the security and provide a transparent and auditable platform. In addition, has the potential to decrease the abstinence in the election, especially for remote votes and save costs, eliminating the need for high costs related to ballot printing, electronic voting machines and maintenance.

## 6.1. Results Obtained

The objectives of this dissertation can be divided into two parts, firstly to identify and explain how the Blockchain technology works and its major properties, explored the technology ecosystem and describe the technology main limitations and challenges. Secondly, the development of a proof of concept in one of the available Blockchain platforms using a use case sufficiently demonstrative to prove the potential advantages of a Blockchain solution.

Both objectives were successfully achieved, in chapters two, three, and four, it is described all information required for the first objective. In the case of the second objective, in chapter five, is presented the developed solution, specifying every component and explaining how it was developed.

Regarding scientific contributions is important to mention that from this dissertation resulted in a paper accepted and published on the Digital Science 2018 conference.

## 6.2. Problems Found

Since Blockchain is a new technology, the available scientific information is still scarce and dispersed, making the research process a more demanding task. To make the situation even worse, due to the increased popularity and media attention to the technology, many forums and articles were written without proper research and fundamentals. Making more difficult to judge if the information presented is accepted as true by the Blockchain community or if it is an unsupported theory.

The process of defining the right approach for the DApp system was also very challenging. Two different approaches were initially developed but the result was not considered a good solution to the electronic voting problem. The first one was a simple voting DApp where the votes were exposed immediately after being cast, this was not considered a good solution because it lacks privacy and could influence the remaining voters. The second approach was based on a commitment scheme, the act of voting was divided into two steps. The vote is first committed and after the deadline of the election, the vote is revealed. This approach was not considered a good solution because the necessity of forcing the voter to reveal his vote to be counted could become problematic in a larger scale application.

The final approach developed includes three contracts that define how the user can interact with the same. The fact that the three contracts also interact with each other became a challenging issue to solve, because the Election contract is created by the Creator contract, and it was not possible to create the new contract because its parameters were not stored on the Blockchain because that contract hasn't been deployed yet. The solution was to deploy an initial standard Election contract, this way its parameters are already stored on the Blockchain and enabling new Election contracts to be spawned.

## 6.3. Future Work

Although all the objectives and expected outcomes of this dissertation were achieved, there still are a few issues and that could be improved in the future.

The breadth of the research topics can be considered vast but the depth of the study is still shallow. Given that, one recommendation to future work is to go deeper and contribute to the research of those topics.

Regarding the developed DApp, there are three relevant improvements to consider in the future. The first one is to deploy the application on the Ethereum main network. Currently, the DApp was deployed on the localhost network, that allows the DApp to be tested and interact with a personal Ethereum Blockchain and accounts. When deployed in the mainnet, the DApp can be used by every Ethereum client, allowing to collect data about the application and users feedback. The downside is that each transaction will cost startgaz, requiring the project to be funded.

One possible disadvantage of using Blockchain as an electronic voting system is that each voter has to pay the startgaz necessary for their vote be transacted. One possible solution to solve this problem is to create a contract, whose owner would be the creator of the election, that would distribute the foreseen amount of ether required for all necessary transaction by every voter. This way is possible to eliminate the inconvenient of each voter having to pay for his vote, and it would allow a new form of verification for an address that is allowed to vote.

Lastly, in this system, it is expected that the permission to create new ballots is to be granted by the administrator, although, in order to facilitate the development and test of the application, that permission is granted automatically by the system. This is one of the not implemented functionalities that would improve the performance of the application.

# REFERENCES

52 INSIGHTS: Don Tapscott. (2018). Retrieved November 6, 2018, from https://www.52-insights.com/don-tapscott-blockchain-represents-the-second-era-of-the-internet-interview/

Baliga, A. (2017). Understanding Blockchain Consensus Models. *Whitepaper*. Persistent Systems Ltd. Retrieved from https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf

Blockchain Research Lab. (n.d.). Retrieved from http://www.blockchainresearchlab.org/projects/retail-and-consumer-goods/

Blockgeeks. (n.d.). Retrieved November 8, 2018, from https://blockgeeks.com/guides/what-is-hashing/#comments

Buterin Vitalik. (2015). On Public and Private Blockchains. Retrieved October 10, 2018, from https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, *4*, 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339

ConsenSys. (n.d.). Dapp Architecture Designs. Retrieved October 23, 2018, from https://github.com/ConsenSys/Ethereum-Development-Best-Practices/wiki/Dapp-Architecture-Designs

Contri, B., & Galaski, R. (2016). Over the horizon Blockchain and the future of financial infrastructure, 21. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/gx-fsi-blockchain-deloitte-summary.pdf

Dagher, G. G., Marella, P. B., Milojkovic, M., & Mohler, J. (2018). BroncoVote: Secure Voting System using Ethereum's Blockchain. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, (Icissp), 96–107. https://doi.org/10.5220/0006609700960107

Deloitte. (2017). Blockchain @ Media - A new Game Changer for the Media Industry? *Monitor Deloitte*, 1–22. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/Deloitte_PoV_Blockchain @ Media.pdf

Dhaou, S. I. Ben. (2018). " Everything and its opposite " Socio-economic implications of Blockchain technology.

Dieterich, V., Ivanovic, M., Meier, T., Zäpfel, S., Utz, M., & Sandner, P. (2017). Application of Blockchain Technology in the Manufacturing Industry, (November), 1–23. Retrieved from www.twitter.com/fsblockchain%0Awww.facebook.de/fsblockchain

E-learning Spot. (n.d.). Retrieved October 11, 2018, from http://learningspot.altervista.org/hash-pointers-and-data-structures/

Ethereum. (n.d.). White Paper. Retrieved November 8, 2018, from https://github.com/ethereum/wiki/wiki/White-Paper

Gregory; Dapp University. (n.d.). The Ultimate Ethereum Dapp Tutorial. Retrieved September 15, 2018, from http://www.dappuniversity.com/articles/the-ultimate-ethereum-dapp-tutorial

Hasse, F., von Perfall, A., Hillebrand, T., Smole, E., Lay, L., & Charlet, M. (2016). Blockchain – an opportunity for energy producers and consumers? *PwC Global Power & Utilities*, 1–45. Retrieved from www.pwc.com/utilities

Jessica, B., Duncan, T., & John, P. (2017). Blockchain: Background, challenges and legal issues. Retrieved from https://www.dlapiper.com/~/media/Files/Insights/Publications/2017/06/Blockchain_background_challenges_legal_issues_V6.pdf

Jiang, S., Cao, J., Wu, H., Yang, Y., Ma, M., & He, J. (2018). BlocHIE: A BLOCkchain-Based Platform for Healthcare Information Exchange. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 49–56). IEEE. https://doi.org/10.1109/SMARTCOMP.2018.00073

Karl Floersch. (2016). Commit-Reveal Voting. Retrieved November 10, 2018, from file:///C:/Users/A541U/Dropbox/Dissertação/Repositorio/dApp/Learning Solidity Part 2 Commit-Reveal Voting.html

Kohno, T., & Stubblefield, A. (2004). Analysis of an electronic voting system. *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium On*, (May), 27–40. https://doi.org/10.1109/SECPRI.2004.1301313

Krawiec, R. J., Housman, D., White, M., Filipova, M., Quarre, F., Barr, D., … Israel, A. (2016). Blockchain: Opportunities for health care. *Proc. NIST Workshop Blockchain Healthcare*, 1–16. Retrieved from https://www2.deloitte.com/us/en/pages/public-sector/articles/blockchain-opportunities-for-health-care.html%0Ahttps://www2.deloitte.com/content/dam/Deloitte/us/Documents/public-sector/us-blockchain-opportunities-for-health-care.pdf

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489–504. https://doi.org/10.1057/ejis.2008.40

Lewis, A. (2017). What's the difference between a distributed ledger and a blockchain? Retrieved from https://bitsonblocks.net/2017/02/20/whats-the-difference-between-a-distributed-ledger-and-a-blockchain/

Lukas Kolisko. (2018). In-depth on differences between public, private and permissioned blockchains. Retrieved November 7, 2018, from https://medium.com/@lkolisko/in-depth-on-differences-between-public-private-and-permissioned-blockchains-aff762f0ca24

McCorry, P., Shahandashti, S. F., & Hao, F. (2017). A smart contract for boardroom voting with maximum voter privacy. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10322 LNCS*, 357–375. https://doi.org/10.1007/978-3-319-70972-7_20

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, 9. https://doi.org/10.1007/s10838-008-9062-0

Ofir Beigel. (2018). What is Double Spending? Retrieved November 7, 2018, from https://99bitcoins.com/double-spending/

Ouattara, H. F., Ahmat, D., Ouédraogo, F. T., Bissyandé, T. F., & Sié, O. (2018). Blockchain Consensus Protocols. In A. 2017 (Ed.), *e-Infrastructure and e-Services for Developing Countries* (pp. 304–314). https://doi.org/10.1007/978-3-319-98827-6_29

Palas Nogueira, J., & De Sá-Soares, F. (2012). Trust in e-voting systems: A case study. In *Lecture Notes in Business Information Processing* (Vol. 129 LNBIP, pp. 51–66). https://doi.org/10.1007/978-3-642-33244-9_4

Parizi, R. M., Amritraj, & Dehghantanha, A. (2018). Smart Contract Programming Languages on Blockchains: An Empirical Evaluation of Usability and Security (pp. 75–91). https://doi.org/10.1007/978-3-319-94478-4_6

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Prisco, G. (2016). Department of Homeland Security Awards Blockchain Tech Development Grants for Identity Management and Privacy Protection. Retrieved November 8, 2018, from https://bitcoinmagazine.com/articles/department-of-homeland-security-awards-blockchain-tech-development-grants-for-identity-management-and-privacy-protection-1471551442/

Swan, M. (2015). Blockchain. In T. McGovern (Ed.), *Blueprint for a New Economy*. O'Reilly Media, Inc.

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*, *2*(9). https://doi.org/10.5210/fm.v2i9.548

Tapscott, D., & Tapscott, A. (2017). The Blockchain Revolution and Higher Education. *EDUCAUSE Review*, 10–24. https://doi.org/10.1038/308683b0

Tapscott, D., & Tapscott, La. (2016). Blockchain and the CIO : a new model for IT. Retrieved November 8, 2018, from https://www.linkedin.com/pulse/blockchain-cio-new-model-don-tapscott/

Vitalik Buterin. (2017). The Meaning of Decentralization. Retrieved November 10, 2018, from https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274

Vom Brocke, J., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Brocke, J., ... Cleven, A. (2009). Association for Information Systems AIS Electronic Library (AISeL) RECONSTRUCTING THE GIANT: ON THE IMPORTANCE OF RIGOUR IN DOCUMENTING THE LITERATURE SEARCH PROCESS Recommended Citation &quot;RECONSTRUCTING THE GIANT: ON THE IMPORTANCE OF RIGOUR IN DOCUM, 1–14. https://doi.org/10.1108/09600031211269721

Voshmgir, S., & Kalinov, V. (2017). Blockchain - A Beginners Guide. Retrieved from https://blockchainhub.net/blockchain-technology/

White, M., Killmeyer, J., & Chew, B. (2017). Will blockchain transform the public sector? Blockchain basics for government. Retrieved from https://dupress.deloitte.com/dup-us-en/industry/public-sector/understanding-basics-of-blockchain-in-government.html

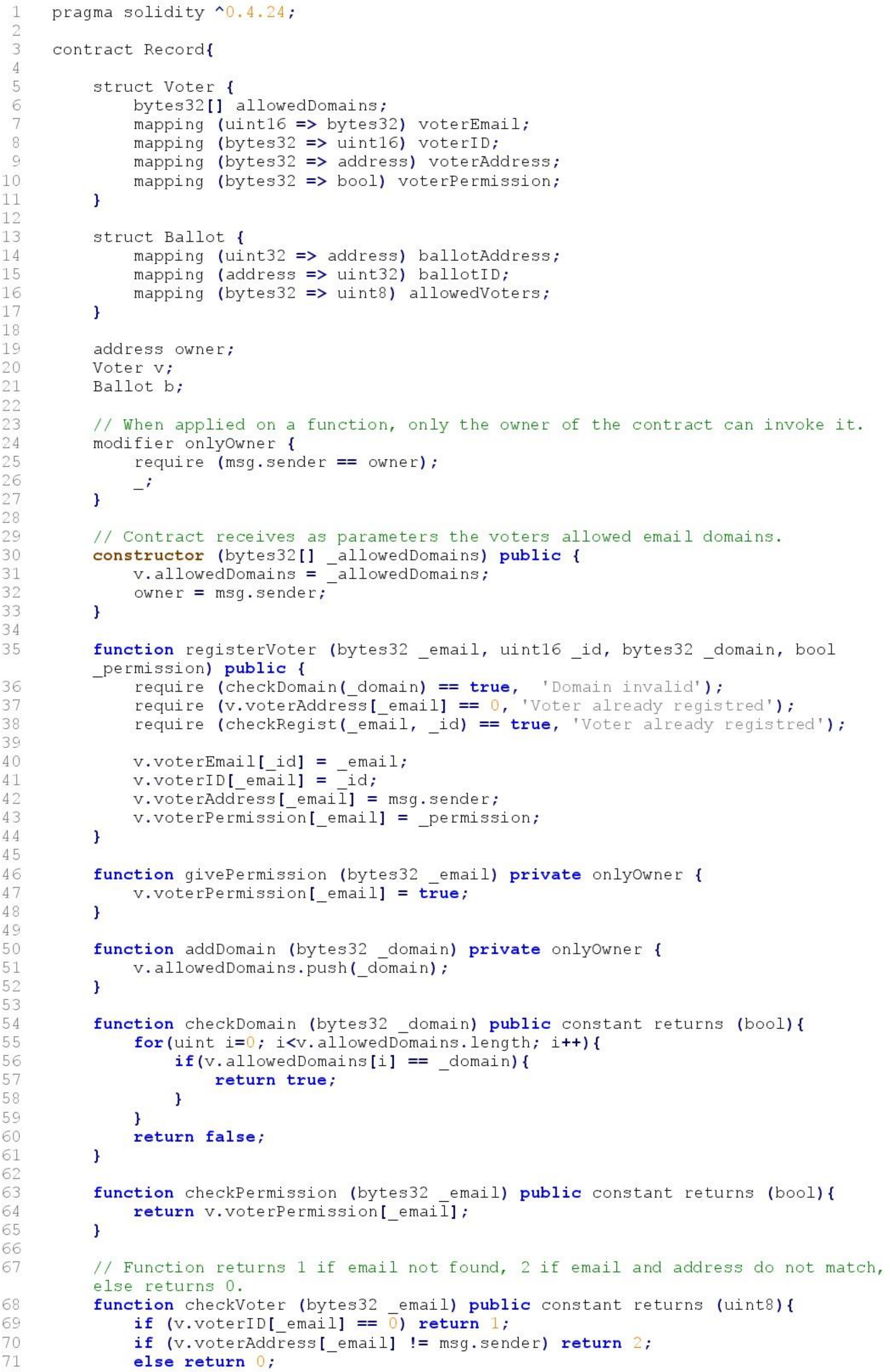
Webster, J., & Watson, R. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly, 26(2), Xiii-Xxiii. Retrieved from http://www.jstor.org/stable/4132319

Yi, X., Paulet, R., & Bertino, E. (2014). *Homomorphic Encryption and Applications*. https://doi.org/10.1007/978-3-319-12229-8

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on Blockchain technology? - A systematic review. *PLoS ONE*, *11*(10), 1–27. https://doi.org/10.1371/journal.pone.0163477

# APPENDICES

# APPENDIX A - RECORD CONTRACT

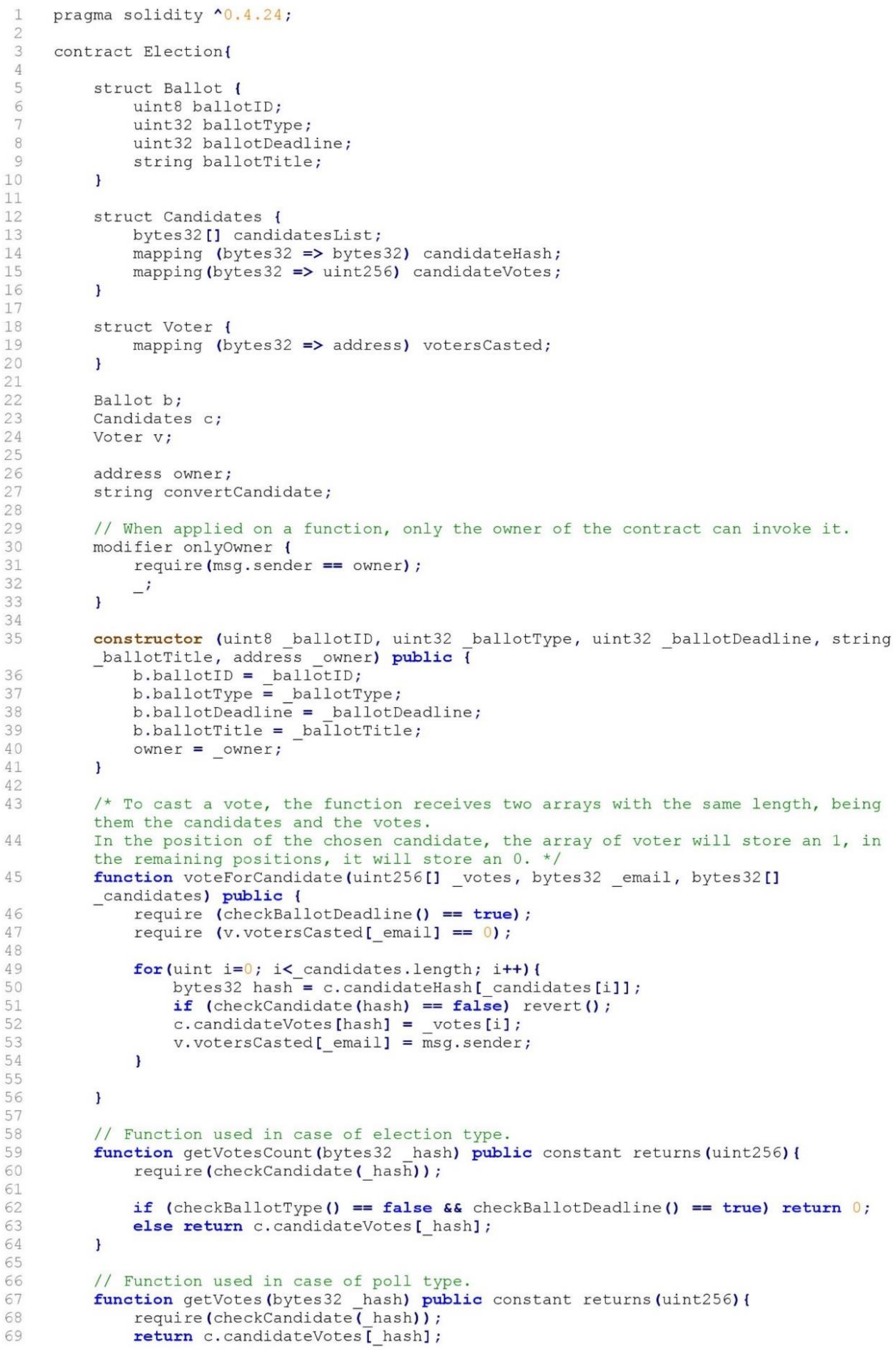


```
pragma solidity ^0.4.24;

contract Record{

    struct Voter {
        bytes32[] allowedDomains;
        mapping (uint16 => bytes32) voterEmail;
        mapping (bytes32 => uint16) voterID;
        mapping (bytes32 => address) voterAddress;
        mapping (bytes32 => bool) voterPermission;
    }

    struct Ballot {
        mapping (uint32 => address) ballotAddress;
        mapping (address => uint32) ballotID;
        mapping (bytes32 => uint8) allowedVoters;
    }

    address owner;
    Voter v;
    Ballot b;

    // When applied on a function, only the owner of the contract can invoke it.
    modifier onlyOwner {
        require (msg.sender == owner);
        _;
    }

    // Contract receives as parameters the voters allowed email domains.
    constructor (bytes32[] _allowedDomains) public {
        v.allowedDomains = _allowedDomains;
        owner = msg.sender;
    }

    function registerVoter (bytes32 _email, uint16 _id, bytes32 _domain, bool _permission) public {
        require (checkDomain(_domain) == true,  'Domain invalid');
        require (v.voterAddress[_email] == 0, 'Voter already registred');
        require (checkRegist(_email, _id) == true, 'Voter already registred');

        v.voterEmail[_id] = _email;
        v.voterID[_email] = _id;
        v.voterAddress[_email] = msg.sender;
        v.voterPermission[_email] = _permission;
    }

    function givePermission (bytes32 _email) private onlyOwner {
        v.voterPermission[_email] = true;
    }

    function addDomain (bytes32 _domain) private onlyOwner {
        v.allowedDomains.push(_domain);
    }

    function checkDomain (bytes32 _domain) public constant returns (bool){
        for(uint i=0; i<v.allowedDomains.length; i++){
            if(v.allowedDomains[i] == _domain){
                return true;
            }
        }
        return false;
    }

    function checkPermission (bytes32 _email) public constant returns (bool){
        return v.voterPermission[_email];
    }

    // Function returns 1 if email not found, 2 if email and address do not match, else returns 0.
    function checkVoter (bytes32 _email) public constant returns (uint8){
        if (v.voterID[_email] == 0) return 1;
        if (v.voterAddress[_email] != msg.sender) return 2;
        else return 0;
    }

    // Check if the user wasn't already registered.
    function checkRegist (bytes32 _email, uint16 _id) public constant returns (bool){
        if (v.voterID[_email] == 0 && v.voterEmail[_id] == 0) return true;
        else return false;
    }

    function getBallotAddress (uint32 _ballotID) public constant returns (address){
        return b.ballotAddress[_ballotID];
    }

    function setBallotAddress (address _ballotAddress, uint32 _ballotID) private {
        b.ballotAddress[_ballotID] = _ballotAddress;
        b.ballotID[_ballotAddress] = _ballotID;
    }
}
```

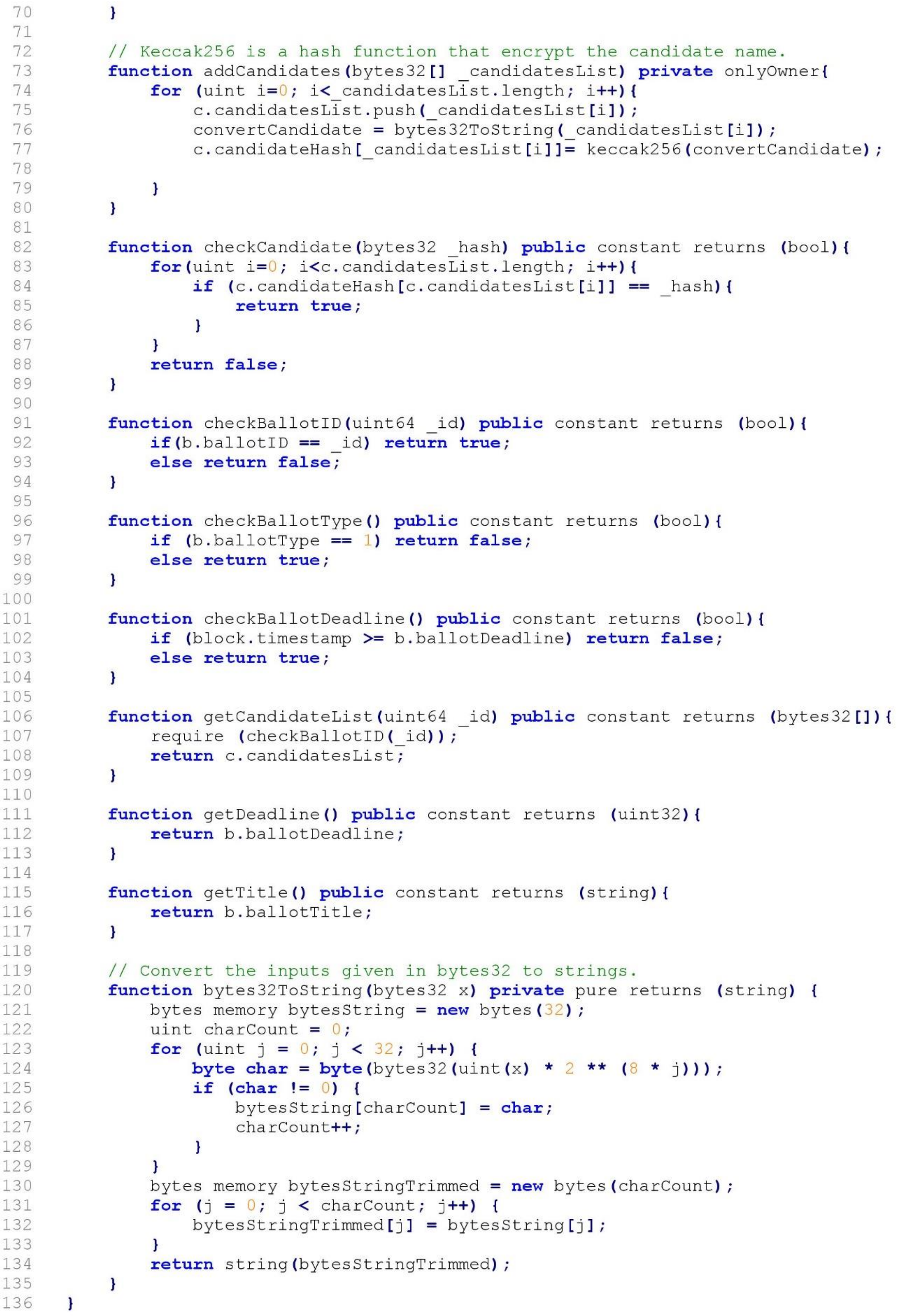
# APPENDIX B - CREATOR CONTRACT

```
pragma solidity ^0.4.24;

import "./Election.sol";

contract Creator{

    mapping (uint32 =>address) ballotAddress;
    address owner;

    // Function receives the required parameters to create an Election contract.
    function createBallot(uint8 _ballotID, uint32 _ballotType, uint32 _ballotDeadline, string _ballotTitle) public {
        owner = msg.sender;
        address newBallot = new Election(_ballotID, _ballotType, _ballotDeadline, _ballotTitle, owner);
        ballotAddress[_ballotID] = newBallot;
    }

    function getAddress (uint32 _ballotID) public constant returns (address){
        return ballotAddress[_ballotID];
    }
}
```

# APPENDIX C - ELECTION CONTRACT



```
pragma solidity ^0.4.24;

contract Election{

    struct Ballot {
        uint8 ballotID;
        uint32 ballotType;
        uint32 ballotDeadline;
        string ballotTitle;
    }

    struct Candidates {
        bytes32[] candidatesList;
        mapping (bytes32 => bytes32) candidateHash;
        mapping(bytes32 => uint256) candidateVotes;
    }

    struct Voter {
        mapping (bytes32 => address) votersCasted;
    }

    Ballot b;
    Candidates c;
    Voter v;

    address owner;
    string convertCandidate;

    // When applied on a function, only the owner of the contract can invoke it.
    modifier onlyOwner {
        require(msg.sender == owner);
        _;
    }

    constructor (uint8 _ballotID, uint32 _ballotType, uint32 _ballotDeadline, string _ballotTitle, address _owner) public {
        b.ballotID = _ballotID;
        b.ballotType = _ballotType;
        b.ballotDeadline = _ballotDeadline;
        b.ballotTitle = _ballotTitle;
        owner = _owner;
    }

    /* To cast a vote, the function receives two arrays with the same length, being them the candidates and the votes.
    In the position of the chosen candidate, the array of voter will store an 1, in the remaining positions, it will store an 0. */
    function voteForCandidate(uint256[] _votes, bytes32 _email, bytes32[] _candidates) public {
        require (checkBallotDeadline() == true);
        require (v.votersCasted[_email] == 0);

        for(uint i=0; i<_candidates.length; i++){
            bytes32 hash = c.candidateHash[_candidates[i]];
            if (checkCandidate(hash) == false) revert();
            c.candidateVotes[hash] = _votes[i];
            v.votersCasted[_email] = msg.sender;
        }

    }

    // Function used in case of election type.
    function getVotesCount(bytes32 _hash) public constant returns(uint256){
        require(checkCandidate(_hash));

        if (checkBallotType() == false && checkBallotDeadline() == true) return 0;
        else return c.candidateVotes[_hash];
    }

    // Function used in case of poll type.
    function getVotes(bytes32 _hash) public constant returns(uint256){
        require(checkCandidate(_hash));
        return c.candidateVotes[_hash];
```



```
    }

    // Keccak256 is a hash function that encrypt the candidate name.
    function addCandidates(bytes32[] _candidatesList) private onlyOwner{
        for (uint i=0; i<_candidatesList.length; i++){
            c.candidatesList.push(_candidatesList[i]);
            convertCandidate = bytes32ToString(_candidatesList[i]);
            c.candidateHash[_candidatesList[i]]= keccak256(convertCandidate);

        }
    }

    function checkCandidate(bytes32 _hash) public constant returns (bool){
        for(uint i=0; i<c.candidatesList.length; i++){
            if (c.candidateHash[c.candidatesList[i]] == _hash){
                return true;
            }
        }
        return false;
    }

    function checkBallotID(uint64 _id) public constant returns (bool){
        if(b.ballotID == _id) return true;
        else return false;
    }

    function checkBallotType() public constant returns (bool){
        if (b.ballotType == 1) return false;
        else return true;
    }

    function checkBallotDeadline() public constant returns (bool){
        if (block.timestamp >= b.ballotDeadline) return false;
        else return true;
    }

    function getCandidateList(uint64 _id) public constant returns (bytes32[]){
        require (checkBallotID(_id));
        return c.candidatesList;
    }

    function getDeadline() public constant returns (uint32){
        return b.ballotDeadline;
    }

    function getTitle() public constant returns (string){
        return b.ballotTitle;
    }

    // Convert the inputs given in bytes32 to strings.
    function bytes32ToString(bytes32 x) private pure returns (string) {
        bytes memory bytesString = new bytes(32);
        uint charCount = 0;
        for (uint j = 0; j < 32; j++) {
            byte char = byte(bytes32(uint(x) * 2 ** (8 * j)));
            if (char != 0) {
                bytesString[charCount] = char;
                charCount++;
            }
        }
        bytes memory bytesStringTrimmed = new bytes(charCount);
        for (j = 0; j < charCount; j++) {
            bytesStringTrimmed[j] = bytesString[j];
        }
        return string(bytesStringTrimmed);
    }
}
```

## APPENDIX D - INTERFACE

```
<!DOCTYPE html>
<html>

<head>
    <title>Voting dApp</title>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700' rel='stylesheet' type='text/css'>
    <link href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css' rel='stylesheet' type='text/css'>
    <link href='stylesheets/app.css' rel='stylesheet' type='text/css'>
</head>

<body class="container" style="background-color:rgb(255,255,255) ">

    <div class = "row" >
        <nav class="navbar navbar-default">
          <div class="container-fluid ">
            <div class="navbar-header">
              <a class="navbar-brand" href="#">
                <img alt="Brand" src="stylesheets/EENG.jpg">
              </a>
            </div>
          </div>
        </nav>
    </div>

    <div class="row">
        <div class="col-sm-5">
            <!-- Load Ballot -->
            <br>
            <h2>Ballot:</h2>
            <br>
            <input type="text" id="ballot_ID" placeholder="Enter ballot ID" style="width: 250px" />
            <a href="#" onclick="loadBallot()" class="button" >Load</a>
            <br>
            <!-- Cast Vote -->
            <br>
            <h2>Vote for your choice:</h2>
            <br>
            <input type="text" id="_email" placeholder="Enter your registered e-mail" style="width: 400px" />
            <br>
            <input type="text" id="candidate" placeholder="Enter the candidate name" style="width: 250px" />
            <a href="#" onclick="castVote()" class="button">Vote</a>
        </div>
        <div class="col-sm-7">
            <!-- Election Results -->
            <br>
            <h2 id="btitle" > Title</h2>
            <div class="table-responsive">
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th style="background-color:rgb(255,255,255)">Candidates</th>
                            <th style="background-color:rgb(255,255,255)">Votes</th>
                        </tr>
                    </thead>
                    <tbody id="candidate-rows" style="background-color:rgb(255,255,255)">
                    </tbody>
                </table>
            </div>
        </div>
    </div>

    <div></div> <hr />

    <div class ="row">
```

```
        <div class="col-sm-5" >
            <!-- Register to Vote -->
            <h2>Register to Vote:</h2>
            <br>
            <input type="text" id="voterEmail" placeholder="Enter your institutional e-mail" style="width: 400px" />
            <br>
            <input type="text" id="voterID" placeholder="Enter your institutional number" style="width: 250px" />
            <br>
            <input type="radio" style="width: 25px" name="voterPermission" value="1"> <b>Request for Ballot Creation Permission</b>
            <br>
            <a href="#" onclick="registerVoter()" class="button">Register</a>
        </div>
        <div class="col-sm-7">
        <!-- Create Ballot -->
            <h2>Create Ballot:</h2>
            <br>
            <input type="text" id="voter_Email" placeholder="Enter your registered e-mail" style="width: 400px" />
            <br>
            <input type="text" id="ballotTitle" placeholder="Enter title of the new ballot" style="width: 400px" />
            <br>
            <b>Candidates name, please seperate each candidate with a comma:</b>
            <br>
            <input type="text" id="ballotCandidates" placeholder="Candidates name" style="width: 400px" />
            <br>
            <div class="row">
                <div class="col-sm-3">
                    <b>Select ballot type:</b>
                    <br>
                    <input type="radio" style="width: 25px" name="ballottype" value="0"> <b>Poll</b>
                    <br>
                    <input type="radio" style="width: 25px" name="ballottype" value="1"> <b>Election</b>
                    <br>
                </div>
                <div class="col-sm-5">
                    <form>
                        <b>Select Poll End Date and Time:</b>
                        <br>
                        <input type="date" id="date" style="width: 170px">
                        <input type="time" id="time" style="width: 125px">
                    </form>
                    <br>
                </div>
            </div>
            <a href="#" onclick="createBallot()" class="button" >Create</a>
        </div>
    </div>
</body>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://cdn.rawgit.com/ethereum/web3.js/develop/dist/web3.js"></script>
<script src="https://cdn.rawgit.com/jiggzson/b5f489af9ad931e3d186/raw/3a316e5b2fa85b8064dd3870936a6162b4862f1f/scientificToDecimal.js"></script>
<script src="https://cdn.rawgit.com/abritinthebay/datejs/master/build/production/date.min.js"></script>
<script src="javascripts/app.js"></script>

</html>
```

# APPENDIX E - API

```
import '../stylesheets/app.css'

import {
    default as Web3
} from 'web3'
import {
    default as contract
} from 'truffle-contract'
import {
    sha3withsize
} from 'solidity-sha3'
import {
    default as HookedWeb3Provider
} from 'hooked-web3-provider'
import {
    default as lightwallet
} from 'eth-lightwallet'


import registrar_artifacts from '../../build/contracts/Registrar.json'
import voting_artifacts from '../../build/contracts/Voting.json'
import creator_artifacts from '../../build/contracts/Creator.json'


var Registrar = contract(registrar_artifacts)
var Voting = contract(voting_artifacts)
var Creator = contract(creator_artifacts)

var ballotID
let candidates = {}

//Set Web3 on page load
$(document).ready(function() {

    if (typeof web3 !== "undefined") {
        window.web3 = new Web3(web3.currentProvider)
    } else {
        window.web3 = new Web3(new
        Web3.providers.HttpProvider("http://localhost:8545"))
    }

    Registrar.setProvider(web3.currentProvider)
    Voting.setProvider(web3.currentProvider)
    Creator.setProvider(web3.currentProvider)

})
//End page load setup


// Register voter
window.registerVoter = function(){
    let voterID = $("#voterID").val()
    let voterEmail = $("#voterEmail").val()
    let voterPermission = $("input[name=voterPermission]:checked").val()
    var domain = voterEmail.replace(/.*@/,"")

    Record.deployed().then(function(contract){
        contract.checkDomain.call(domain).then(function(_domain){
            var validDomain = _domain.toString()
            if (validDomain == "false"){
                window.alert("Invalid email domain.")
                throw new Error()
            }

            contract.checkRegist.call(voterEmail, voterID).then(function(_regist){
                var validRegist = _regist.toString()
                if(validRegist == "false"){
                    window.alert("Voter already registered.")
                    throw new Error()
                }

                // 2500000 it's estimative of the required startgaz for the
                transaction, that amount will be withdrawal from first Ganache
                address.
                contract.registerVoter(voterEmail, voterID, domain, voterPermission, {
                    gas: 2500000,
                    from: web3.eth.accounts[0]
                }).then(function(){
                    window.alert("Voter successfully registered.")
                })
                })
            })
        }
    }
}
// END Register voter

// Create Ballot
window.createBallot = function(){
    let voter_Email = $("#voter_Email").val()

    Record.deployed().then(function(contract){
        contract.checkVoter.call(voter_Email).then(function(_voter){
            var response = _voter.toString()
            if(response == 1){
                window.alert("Email not registered.")
                throw new Error()
            }else if(response == 2){
                window.alert("Email and Ethereum address mismatch!")
                throw new Error()
            }else{

                contract.checkPermission.call(voter_Email).then(function(_permission){
                    let permission = _permission.toString()
                    if(permission == "false"){
                        window.alert("Voter not allowed to creat ballots.")
                        throw new Error()
                    }else{
                        let _date = $("#date").val()
                        let _time = $("#time").val()
                        var timeArray = _time.split(':')
                        var seconds = ((timeArray[0]*60)*60) + (timeArray[1]*60)

                        /*  If using testNet uncomment, it increases 7 hours
                            var seconds = ((timeArray[0]*60)*60) + (timeArray[1]*60)
                            + 21600 */

                       var deadline = (Date.parse(_date).getTime() / 1000)
                        deadline += seconds
                        let ballotType = $("input[name=ballotType]:checked").val()
                        let ballotTitle = $("#ballotTitle").val()
                        let ballotCandidates = $("#ballotCandidates").val()
                        var candidatesArray = ballotCandidates.split(/¥s*,¥s*/)
                        let ballotID = Math.floor(Math.random() * 4294967295)

                        Creator.deployed()then(function(contract){
                            contract.createBallot(ballotID, ballotType, deadline,
                            ballotTitle,{
                                gas: 2500000
                                from: web3.eth.accounts[0]
                            }).then(function(){

                                contract.getAddress.call(ballotID).then(function(_addr
                                ess){
                                    var address = _address.toString()
                                    fillSetup(address, candidatesArray, ballotID)
                                    registerBallot(address, ballotID)
                                    window.alert(address)
                                })
                            })
                        })
                    }
                })
            }
        })
    })
```

```
}

function registerBallot(address, ballotID){
    Record.deployed().then(function(contract){
        contract.setBallotAddress(address, ballotID,{
            gas: 2500000
            from: web3.eth.accounts[0]
        }).then(function(){
            window.alert("Ballot created with success. Please save Ballot ID: "+ ballotID)}
    })
})

function fillSetup(address, candidatesArray, ballotID){
    Election.at(address).then(function(contract){
        contract.addCandidates(candidatesArray,{
            gas: 2500000
            from: web3.eth.accounts[0]
        }
    })
}
// END Create Ballot

//Load Ballot
window.loadBallot = function(){
    $("#candidate-rows tr").remove()
    ballot_ID = $("#ballot_ID").val()

    Record.deployed().then(function(contract){
        contract.getBallotAddress.call(ballot_ID).then(function(_addr){
            var _ballotAddress = _addr.toString();
            if(_ballotAddress == 0){
                window.alert("Invalid Ballot ID.")
                throw new Error()
            }else{
                getCandidates(_ballotAddress ballot_ID)
            }
        })

    })
}
//END Load Ballot

//Vote
window.castVote = function(candidate){
    let _candidateName = $("#candidate").val()
    let _email = $("#_email").val()
    var _Domain = _email.replace(/.*@/,"")
    var candidateHash = sha3withsize(_candidateName, 32)
    var votesArray = []

    Record.deployed().then(function(contract){
        contract.checkVoter(_email,{
            gas: 2500000
            from: web3.eth.accounts[0]
        }).then(function(_voter){
            var response = _voter.toString()
            if(response == 1){
                window.alert("Email not registered.")
                throw new Error()
            }else if(response == 2){
                window.alert("Email and Ethereum address mismatch!")
                throw new Error()
            }

            contract.getAddress.call(ballotID).then(function(_address){
                var ballotAddress = _address.toString();
                Election.at(ballotAddress).then(function(contract){

                    contract.checkCandidate.call(candidateHash).then(function(validCand){
                        var _candidate = validCand.toString()
                        if(_candidate == "false"){
                            window.alert("Invalid candidate.")
                            throw new Error()
                        }

                        /* Function will encrypt two arrays with the same length, being them the candidates and the votes.
                           In the position of the chosen candidate, the array of voter will store an 1, in the remaining positions, it will store an 0. */
                        contract.getCandidateList.call(ballotID).then(function(_candidatesArray){
                            for(let i=0; i<_candidatesArray.length; i++){
                                let _cand = (web3.toUtf8(_candidatesArray[i]))
                                let _candidateHash = sha3withsize(_cand, 32)
                                if(candidateHash == _candidateHash){
                                    encrypt(_candidateHash, 1, i, candidatesArray, _email, ballotAddress, votesArray)
                                }else{
                                    encrypt(_candidateHash, 0, i, candidatesArray, _email, ballotAddress, votesArray)
                                }
                            }
                        })
                    })
                })
            })
        }
    }
}

// Function will send the casted vote to the encryption server to be encrypted.
function encrypt (_candidateHash, _vote, i, candidatesArray, _email, ballotAddress, votesArray){
    var voteNum
    $.ajax({
        type: "GET",
        url: "http://localhost:3000/encrypt/" + _vote,
        success: function (_voteEncrypt){
            Election.at(ballotAddress).then(function(contract){
                contract.getVotes.call(_candidateHash).then(function(v){
                    voteNum = v.toString()
                    voteNum = scientificToDecimal(voteNum)
                    if (voteNum != 0){
                        add(_voteEncrypt, voteNum, i, candidatesArray, _email, ballotAddress, votesArray)
                    }
                })
            })
        }
    })
}

// Function will update the encrypted vote count.
function add(_voteEncrypt, voteNum, i, candidatesArray, _email, ballotAddress, votesArray){
    $.ajax({
        type: "GET",
        url: "http://localhost:3000/add/" + _voteEncrypt + "/" + voteNum,
        success: function (voteCountEncrypt){
            checkTimeStamp(voteCountEncrypt, i, candidatesArray, _email, ballotAddress, votesArray)
        }
    })
}

function checkTimeStamp(voteCountEncrypt, i, candidatesArray, _email, ballotAddress, votesArray){
    Election.at(ballotAddress).then(function(contract){
        contract.checkBallotDeadline.call()then.(function(t){
            var timeCheck = t.toString()
            if(timeCheck == "false"){
                contract.getDeadline.call().then(function(t){
                    var _deadline = t.toString()
                    // If using testNet uncomment, it increases 7 hours
                    //_deadline = _deadline - 21600
                    _deadline = new Date (_deadline * 1000)
                    getVoteCount(ballotAddress)
                    window.alert("Deadline ended on " + _deadline)
                    throw new Error()
                })
            } else{
                votesArray[i] = voteCountEncrypt
                if(i==candidatesArray.length - 1){
                    getVote(i, candidatesArray, _email, ballotAddress, votesArray)
                }
            }
        })
    })
}

function getVote(i, candidatesArray, _email, ballotAddress, votesArray){
    Election.at(ballotAddress).then(function(contract){
            gas: 2500000,
            from: web3.eth.accounts[0]
        }).then(function() {
            getVoteCount(ballotAddress)
            window.alert("Your vote has been verified.")
        })
    })
}
//END Vote

//Get Votes
function getCandidates(ballotAddress, ballotID){
    Election.at(ballotAddress).then(function(contract){
        contract.getTitle.call().then(function(_ballotTitle){
            contract.getCandidateList.call(ballotID).then(function(_candidatesArray){
                for(let i=0; i<_candidatesArray.length; i++){
                    candidates[web3.toUtf8(_candidatesArray[i])] = "candidate: " + i
                }
                setupTable()
                getVotes(ballotAddress)
            })
        })
    })
}

// Display votes on table.
function setupTable() {
    Object.keys(candidates).forEach(function(candidate) {
        $("#candidate-rows").append("<tr><td>" + candidate + "</td><td id='" + candidates[candidate] + "'></td></tr>");
    })
}

function getVotes(ballotAddress){
    let candidatesName = Object.keys(candidates)
    for(var i=0; i<candidatesName.length; i++){
        let _name = candidatesName[i]
        let candiHash = sha3withsize(_name, 32)

        Election.at(ballotAddress).then(contract){
            contract.getVotesCount.call(candiHash).then(function(voteNum){
                var _voteNum = voteNum.toString()
                if(_voteNum == 0){
                    contract.getDeadline.call().then(function(t){
                        var endTime = t.toString()
                        // If using testNet uncomment, it increases 7 hours
                        //_deadline = _deadline - 21600
                        endTime = new Date(endTime * 1000);
                        window.alert("Results will be displayed once the voting period has ended (" + endTime + ")")
                    })
                }else{
                    _voteNum = scientificToDecimal(_voteNum)
                    decrypt(_voteNum, _name)
                }
            })
        }
    }
}

// Function will decrypt the vote count.
function decrypt(_voteNum, _name) {
    $.ajax({
        type: "GET",
        url: "http://localhost:3000/decrypt/" + _voteNum,
        success: function(_VoteCount) {
            var VoteCount = _VoteCount
        }
    })
}

```

## APPENDIX F - ENCRYPTION SERVER

```
const http = require('http')

var express = require('express')
var paillier = require('jspaillier')
var jsbn = require('jsbn')
var body = require('body-parser')
require('datejs')

var app = express()

var keys = paillier.generateKeys(128)

const hostname = '127.0.0.1'
const port = 3000

app.get('/', function(req, res) {
    res.send('BroncoVotes: Backend Server')
})

app.listen(port, function(res) {
    console.log('BroncoVotes: Backend Server Listening on Port ' + port)
})

app.use(function(req, res, next) {
    res.header('Access-Control-Allow-Origin', '*')
    res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept')
    next()
})

// Encrypt vote.
app.get('/encrypt/:id', function(req, res) {
    var ekey = req.params.id
    ekey = keys.pub.encrypt(keys.pub.convertToBn(ekey)).toString()
    res.send(ekey)
})

// Decrypt vote count.
app.get('/decrypt/:id', function(req, res) {
    var dkey = req.params.id
    dkey = keys.sec.decrypt(keys.pub.convertToBn(dkey)).toString()
    res.send(dkey)
})

// Update vote count.
app.get('/add/:id/:id2', function(req, res) {
    var ein1 = req.params.id
    var ein2 = req.params.id2
    eadd = keys.pub.add(keys.pub.convertToBn(ein1), keys.pub.convertToBn(ein2)).toString()
    res.send(eadd)
})

app.get('/getTime', function(req, res) {
    var timestamp = Math.round((new Date()).getTime() / 1000)
    res.send("" + timestamp)
})

```