FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Development of a Blockchain Based Energy Trading Scheme for Prosumers

Artur Pinto de Almeida

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: João Paulo da Silva Catalão

Second Supervisor: Matthew Brian Gough

July 6, 2020

© Artur Pinto de Almeida, 2020

Resumo

Ao longo dos últimos anos o mercado energético tem sofrido grandes mudanças devido ao contínuo aumento da procura de energia, à introdução de tecnologias de comunicação ao nível da rede e dos consumidores e à introdução de fontes distribuídas de geração, incluindo em aplicações domésticas tornando os consumidores em prosumers. Desta forma, os sistemas de energia têm se tornado cada vez mais descentralizados através do desenvolvimento digital da *Intenet of Things* (IoT) com vista à criação das *smart grids* colocando também o consumidor no centro do sistema elétrico. Nesta perspetiva, uma tecnologia que deve ser destacada é a *Blockchain*, que tem revelado possuir um enorme potencial no setor económico. A investigação do uso de *Blockchain* no sector elétrico tem vindo a crescer de uma forma contínua sobretudo devido à sua natureza distribuída e imutável que pode potenciar o aceleramento da transição do sistema elétrico, para um sistema cada vez mais digitalizado e descentralizado.

Neste trabalho de dissertação são exploradas as capacidades da *Blockchain*, adaptadas para o desenvolvimento de um esquema de transações de energia para *prosumers*. O método utilizado consiste na criação de um mercado virtual baseado em *smart contracts* da *Blockchain Ethereum*, utilizando a linguagem Solidity em ambiente Remix, ao qual serão adicionados consumidores e produtores de energia que irão comunicar entre si, trocando tokens por eletricidade de modo a satisfazer as necessidades de toda a comunidade envolvente. A utilização de *Blockchain* irá assim possibilitar a criação de processos de vinculação autónomos, permitindo aos seus utilizadores a venda de energia, de forma segura e fiável, sem a necessidade de uma terceira entidade.

Palavras chave: Blockchain, Ethereum, Solidity, Contratos Inteligentes, Mercado de Energia, Smart Grid, Micro-redes.

Abstract

Throughout the last few years, the electric market has undergone countless changes due to the continuous demand growth, the introduction of Information and Communication Technologies (ICT) on the network level, and the introduction of distributed generation sources, including household appliances that turned some consumers into a new category, prosumers. This way, the energy systems have become more decentralized through the development of the Internet of Things (IoT) which can allow for the creation of smart grids and putting the consumer on the center of the electric system. From this aspect, an interesting technology is the blockchain, which has revealed itself to have a huge potential in the financial sector. The research for the use of blockchain in the electric system has been due to its distributed and immutable nature which provides the potential to accelerate the transition of the electric market to a more digital and decentralized system.

In this thesis, the blockchain capacities will be researched and adapted to develop of a scheme of energy transactions on the prosumer level. This will be done through the creation of a virtual market based on Ethereum smart contracts, using the Solidity language in a Remix environment, which will also allow consumers and energy producers to communicate between each other, exchanging tokens for electricity to satisfy the needs of all the evolved community. The use of blockchain will then make it possible for the creation of autonomous binding processes, allowing the system users to exchange energy, in a reliable and secure way, without the need of a third entity.

Keywords: Blockchain, Ethereum, Solidity, Smart Contracts, Electricity Markets, Smart Grid, Microgrids.

Acknowledgements

Gostaria de agradecer ao meu orientador, O Professor João Catalão, pela oportunidade de realizar este trabalho de dissertação com ele e com a sua equipa. Aos meus co-orientadores, Mathew Gough e Sérgio Santos, pelo apoio e dedicação ao longo destes cinco meses de trabalho, sem os quais este projeto não poderia estar a ser entregue.

Um agradecimento à minha família, em especial ao meu pai, mãe, avós, avô e tia, pelo esforço que fizeram não só no decorrer deste projeto como também ao longo de todos os cinco anos do curso. Pelo apoio mais que financeiro mas também moral para que conseguisse estar a estudar em outra cidade querendo me proporcionar algo que nunca eles tiveram.

Aos meus amigos. Àqueles que já se encontravam ao meu lado antes de entrar para a faculdade e aos que entraram para a minha vida durante todo este percurso. Aos mais próximos, à Maria Pereira e à Sara Costa em especial, pela força que me deram e pela ajuda em me lembrar o porque da escolha deste percurso para o meu futuro.

Artur Pinto de Almeida

"The adventure of life is to learn.
The purpose of life is to grow.
The nature of life is to change.
The challenge of life is to overcome.
...
The secret of life is to dare.
..."

William Arthur Ward

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Problem Definition and Motivation . . . 4
- 1.3 Objectives . . . 6
- 1.4 Methodology . . . 6
- 1.5 Dissertation Structure . . . 7

2 Blockchain **9**
- 2.1 History and Present Interest . . . 9
- 2.2 Basic Principles and Architecture . . . 12
- 2.3 Features of the Technology . . . 14
- 2.4 Gas . . . 16
- 2.5 Prospects in the Electric Market . . . 18
  - 2.5.1 Opportunities . . . 19
  - 2.5.2 Challenges . . . 21
- 2.6 Chapter summary . . . 22

3 Software Development **23**
- 3.1 Remix . . . 23
- 3.2 Entities . . . 26
  - 3.2.1 Administrator . . . 26
  - 3.2.2 Consumer . . . 27
  - 3.2.3 Producer . . . 27
  - 3.2.4 Network Coordinator . . . 27
  - 3.2.5 Storing of agent's information . . . 28
- 3.3 Demands and Offers . . . 29
- 3.4 Token . . . 31
  - 3.4.1 Project application . . . 31
  - 3.4.2 Payment safety . . . 32
- 3.5 Contract Structure . . . 33
  - 3.5.1 Administrator contract . . . 34
  - 3.5.2 User contract . . . 34
  - 3.5.3 Market contract . . . 34
  - 3.5.4 Important restrictions . . . 35
- 3.6 Market Design . . . 36
  - 3.6.1 Time blocks - organization . . . 36
  - 3.6.2 Energy price . . . 37
  - 3.6.3 Transfer of energy for token . . . 38

3.7 Gas Usage . . . 39
3.8 Chapter summary . . . 40

4 Testing and Results **41**
4.1 Test Environment . . . 41
4.2 Administrator . . . 42
4.3 User contract . . . 44
4.3.1 Register users . . . 44
4.3.2 Authorize a prosumer . . . 46
4.3.3 Network Coordinator . . . 47
4.4 Market contract . . . 47
4.4.1 Asks . . . 47
4.4.2 Bids . . . 48
4.4.3 Accept an order . . . 49
4.4.4 Clearing of market information . . . 51
4.5 Cost Benefits . . . 52
4.6 Chapter summary . . . 53

5 Conclusions and Future Works **55**
5.1 Conclusions . . . 55
5.2 Future Work . . . 56

References **57**

List of Figures

1.1 3D revolution . . . 3
1.2 Comparing the present market structure to a structure using blockchain technology 4
1.3 Blockchain usage on the energy market . . . 5

2.1 Block creation on PoW consensus protocol [1]. . . . 10
2.2 Difference between permissioned (left) and permissionless (right) blockchains [2]. 13
2.3 Process to create a transaction in blockchain . . . 15
2.4 Gas paid as fee . . . 16
2.5 Use cases of blockchain on the energy sector [3]. . . . 21

3.1 Remix IDE . . . 24
3.2 Manually compiled code . . . 25
3.3 Manually deployed code . . . 25
3.4 System's use case . . . 26
3.5 User Storage . . . 28
3.6 Example of a mapping . . . 29
3.7 Storage of order information . . . 30
3.8 Energy transfer example . . . 32
3.9 System's class diagram . . . 33
3.10 Administrator contract excerpt . . . 34
3.11 Service user contract excerpt . . . 35
3.12 Market contract excerpt . . . 35
3.13 Function's modifiers . . . 36
3.14 Example intersection of curves . . . 38
3.15 Code excerpt for transfer storage . . . 39

4.1 Test window . . . 42
4.2 Successful transaction . . . 43
4.3 Administrator contract . . . 43
4.4 Gas limit reached . . . 44
4.5 User contract deployed . . . 45
4.6 Trying to add a new user by other address than the administrator . . . 45
4.7 Trying to give allowance to a not registered address . . . 46
4.8 Looking into a prosumer structure . . . 46
4.9 Trying to execute an order with admin account . . . 48
4.10 Creating an ask . . . 48
4.11 Trying to delete an ask of another account . . . 48
4.12 Trying to execute an order with a consumer account . . . 49
4.13 Creating a bid . . . 49

4.14 Trying to delete a bid of another account 49
4.15 Trying to accept a bid with a non-user account 50
4.16 Trying to accept an ask with a consumer 50
4.17 Successful acceptance of a bid 51
4.18 Seller information 51
4.19 Buyer information 51

List of Tables

2.1 Gas price chart 17

3.1 Example of storing offers 31
3.2 Example of offers for a time-block 37

4.1 System's tests entities 47
4.2 Executed orders on the market 52
4.3 Traded offers among users and NC 52
4.4 Costs and savings for users 52
4.5 Fees costs 53

Acronyms and Symbols

AI	Artificial Intelligence
API	Application Program Interface
BTC	Bitcoin
DAO	Decentralized Autonomous Organization
DApp	Decentralized Applications
DER	Distributed Energy Resources
DGpr	General Data Protection Regulator
DR	Demand Response
EDP	Energias de Portugal
EU	European Union
ETH	Ether
ERC	Ethereum Request for Comment
EV	Electric Vehicle
EVM	Ethereum Virtual Machine
GHG	Greenhouse Gases
ICO	Initial Coin Offer
ICT	Information and Communication Technologies
IDE	Integrated Development Environment
IoT	Internet of Things
NC	Network Coordinator
P2P	peer-to-peer
POW	Proof of Work
REC	Renewable Energy Credit
RES	Renewable Energy Sources
RPOW	Reusable Proof of Work

Chapter 1

Introduction

This project has as a main foundation the electric market. For this reason it is important to understand in which position it encounters itself in the present situation. On this chapter the present electric system will be described, the changes it is facing and the purposes intended to be reached in the future with this revolutionary shift.

By the end of this chapter the reader will better understand the purposes of this thesis, the objectives and how it is organized in order to reach the final objectives.

1.1 Background

The present power system was developed in a centralized way, with few large-scale generator groups, which were usually located far away from highly populated areas and connected to these high demand areas through a complex network of transmission and distribution networks [4]. Nowadays, the system is a outdated structure as it was designed more than one century ago and significant change has occurred in the energy system since then [5]. This centralized structure based on fossil fuels has high costs, high production losses as well as high losses in energy transmission and distribution, mainly in areas far from generating units, which cannot always be placed in an optimal position [6].

The intense technological revolution the world faces is also affecting the energy sector and with it new and more efficient ways of generating [7], [8], transporting, converting [9], [10] and storing [11] has emerged. Not only are new methods more efficient but they are also more environmentally friendly which has become a major priority, especially in the EU [12], [13], [14]. These new approaches also promise to make it possible for the consumer to become more active within the system, as it will be further explained later in this thesis.

According to the World Economic Forum there are some main drivers for the rapid change of the energy sector [15]. The first is the gradual growth in energy consumption. Since the world population is rising and developing countries, for example India, are using more modern carriers of energy, it is expected that the energy consumption will continue to grow significantly in the coming years [16]. This increase also comes with the electrification of various economic sectors,

for example, the automobile sector or residential heating [2]. Also, the continuous integration of renewable sources in large groups of generators or as residential applications has the potential to reshape the current market structure and the direction of energy flow in the grid. The inclusion of small generating units also allows the creation of smaller networks, independent or not from the main power grid, or also called as micro-grids [17]. Lastly, the fast development of digital technologies that ensure a cost-effective way of obtaining data, always taking to account user security and reliability of information, with a touch of luxury.

This rapid reshape of the market can also be referred as the 3 D's revolution (decarbonization, digitalization and decentralization) and this is shown in Figure 1.1 [18]:

- Decarbonization: this is a aim to be reached in the next few decades with new goals imposed every year with the scope to reduce greenhouse gases (GHG) emissions. To reach that objective we should embrace renewables in big generating centers as in domestic or commercial applications [19];

- Digitalization: the purpose of reaching a digitalized sector is to improve the communication between all the involved participants [20]. This way it is possible to know consumer demand and predict the needs of an entire region more easily. This has been introduced in Portugal by companies such as EDP with the installation of smart meters [21] or commercialization of smart plugs;

- Decentralization: this means removing the entity that regulates the distribution market by giving more power to the consumer. The development of more secure technologies, such as blockchain [22], make it easier to adopt a decentralized market system.

These three factors work together and give each other reasons to grow. As soon as a consumer integrates any renewable source for own production, the first step towards decentralization is being made. As more consumers are considered as producers as well, the more power is being taken from big power plant generators to small entities [23].

The integration of Renewable Energy Sources (RES) should be accompanied by digital instruments to ensure the safety of the network [24]. These devices, which allow bidirectional flow of information and improved management of the grid functioning, are an important step to improve efficiency and reduce energy congestion in the smart grid [25]. Smart meters, in residential or commercial applications, also give costumers and utility services the knowledge of the consumption rates which result in a faster response since the profiles can be built for specific users. In fact, the number of internet connected devices is estimated to reach 30 million by the end of 2020 bringing Internet of things to the electric market [26] when it already plays an important role in other sectors of the world economy. By connecting to mobile devices such as smartphones, laptops or smartwatches, Internet of Things (IoT) became a time efficient way to monitor energy consumption and making it cost effective and efficient. The enormous quantity of data collected when compared to the present market comes with a necessity for a more efficient way to process that information and that is why a junction with artificial intelligence (AI) has big benefits [27].

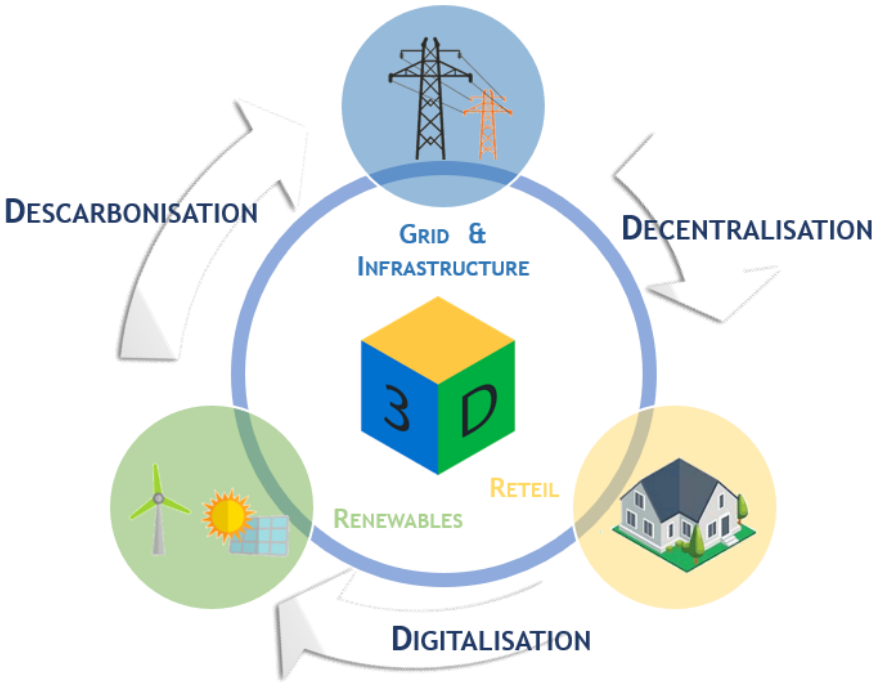


Figure 1.1: 3D revolution

However, the digitalization comes with distrust by some uninformed people that are afraid of seeing their data being accessed by anyone, what makes a good reason to adopt a thrust-worthy technology. Blockchain has been proven to be a secure and reliable way to safely store data [28]. The adoptions of blockchain could make reach another level in the decentralized market by turning consumers into producers who can sell energy to other consumers [29].

The energy sector is currently one of the most advanced sectors in its adoption of blockchain; the technology is perceived by energy leaders to be an issue of both high impact and uncertainty [6]. Numerous blockchain-based projects are emerging within this field and start-ups, incumbent energy companies and national governments are all investigating the potential of the new technology [30], [31].

Although blockchain is a promising and potentially disruptive technology, it is still in an emerging phase, and there is a lack of full scale, commercialized applications [32]. Many uncertainties stand in the way of full technological implementation and they could potentially limit, or even stall, the growth of the technology. As the technology is relatively new and limited knowledge exists about what changes it will bring to the current market structure, it becomes essential to gain further understanding about both the capabilities and limitations of the technology and its use within the energy sector [33].

1.2 Problem Definition and Motivation

As was discussed previously, the structure of the electric system is somewhat outdated and is shown in the left hand side of Figure 1.2. Constructed with the same infrastructures that were designed in a century ago [34] which can have significant losses [35] if it is pretended to have a high efficient distribution system. The market is also constructed according to a monopoly structure [34], where big companies influence the price of electricity paid by consumers so that these losses of generations and transmission can be paid and they can have as much profit as they want. This inflation of prices is due to a much higher number of consumers than producers. The low number of producers let them have full control of the system and let consumers with no other choice than accept their rules if they want to be part of the system [33].

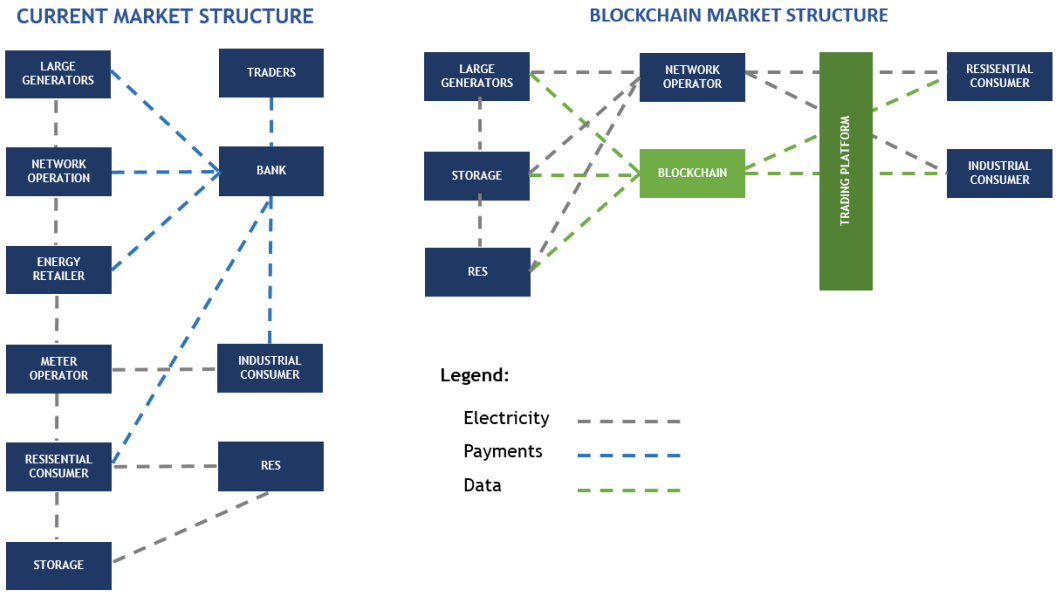


Figure 1.2: Comparing the present market structure to a structure using blockchain technology

The introduction of renewables came with some advantage for consumers since now they can produce energy for their needs and help to reduce the dependence on large utilities. But it is still not enough. The rules in place don't let them have full control of these technologies and some changes need to be made in those legislation's [36] and [37]. The future brings more power to consumers to produce, store and eventually sell their energy to other consumers [38]. Those types of consumers who consume and produce energy are described as prosumers [25].

The growing participation of prosumers with the grid bring along another challenge, the transition that is required in the energy system does not only concern the transition from a centralized system towards a decentralized system, and from grey energy towards green energy, but also a shift in organizational emphasis from top down towards bottom up concerns as can be seen in Figure

1.2 [39], [40]. Recently, there has been a rise in media attention concerning the lack of transparency for customers in their utility bills due to the complexity of those bills [19]. On account of the anticipated rise of RES, the expectation is that these bills will only grow more complex and therefore the need for transparency becomes more pressing. Furthermore, media articles point out that there is little trust among customers in smart meters, as data privacy on smart meters cannot fully be guaranteed [41]. In sum, the existing electricity system is not only challenged by a growing need for flexibility, but is also designed in a top down manner and thereby neglects the role for prosumers by offering no guarantee of data security and little data transparency. The future electricity system is likely to contain a large number of prosumers that want to participate in the electricity system [42]. Therefore, to cope with the ongoing energy transition, a platform is required that can deal with increasing demand-supply fluctuations, stimulate cooperation between prosumers and wholesale market actors and can guarantee data security and transparency for prosumers [41]. This dissertation focuses on one of the pieces of the puzzle, the creation of smart contracts (business rules in a programmable language onto the blockchain and are enforced by the participants of the network), based on blockchain technology, that could cope with these challenges and a potential system of consumer interaction is shown in Figure 1.3. The intended with this project is to develop a smart contract, in Ethereum based energy trading system, facilitate peer-to-peer transactions and allow individuals to take place both as consumers and producers (becoming a prosumer).

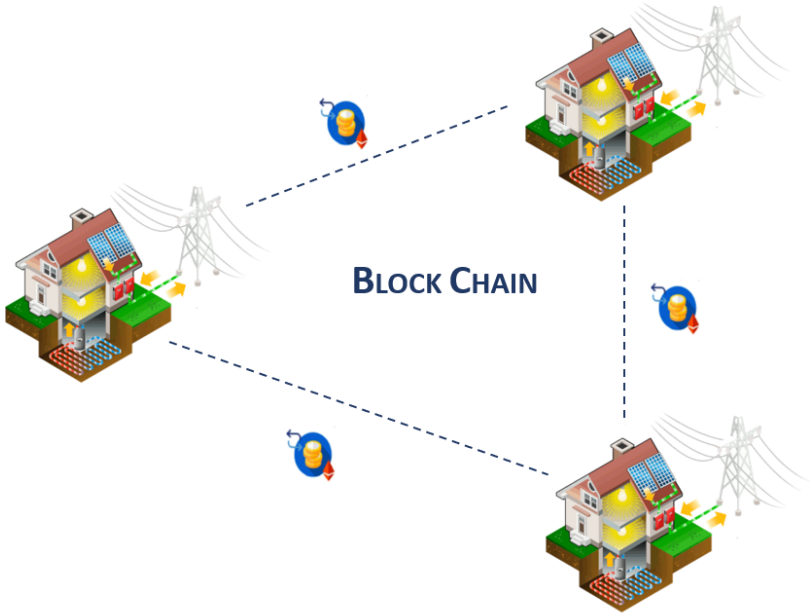


Figure 1.3: Blockchain usage on the energy market

1.3 Objectives

To enable energy trading between two individuals, an agreement of some sort is needed. One party to this agreement takes the obligation of transferring the electric energy to the other who is obligated to pay for that same energy received. This is where blockchain takes the more important role, more specially Ethereum that differentiates itself from the most used blockchain, Bitcoin, by allowing the construction of smart contracts.

Smart contracts represent the agreement made by those two individuals and the obligation they make with each other. Since one is selling energy in the market, and the other wants to buy energy from the market, if the price the seller wants for the energy it is trying to sell and its quantity are not higher than the price the buyer is willing to pay for it, an agreement is reached and the contract is signed by both. The individual capable of supplying the energy is from now on referred as a prosumer, since it takes the place of both consumer and producer of energy.

The main objectives of this dissertation remain in:

- Identify the impact of peer-to-peer (P2P) trading smart contract for:
  - Consumers and prosumers;
  - The value for the supply side;
  - The need for new market players.
- Create a P2P trading smart contract to facilitate peer-to-peer transactions and allow individuals to take place both as consumers and producers (becoming a prosumer).

In short, the intended with this project is to develop a smart contract, in Ethereum based energy trading system, facilitate peer-to-peer transactions and allow individuals to take place both as consumers and producers and become a prosumer. In a final phase, is intended to create a simulation program composed by residential prosumers with different consume needs in quantity or by periods of time. The purpose it to better understand how the network would work and if the system would be able to easily support itself.

1.4 Methodology

The proposed project aims to develop a smart contract, in Ethereum based energy trading system, to facilitate peer-to-peer transactions and allow individuals to act as both consumers and producers (becoming a prosumer) as presented in Figure 1.3. A trading scheme will be built, composed of between 6 to 8 prosumers that belong to the network and sell and buy energy between each other. The smart contract will be developed in Solidity, version 0.6.6. All the simulations are conducted in an Lenovo ideapad 530s workstation with four intel core i7 processors, each clocking at 1.8GHz frequency, and 16 GB of RAM.

1.5 Dissertation Structure

This dissertation is divided in six chapters. Chapter 2 presents the state of art and the related concepts on the topic being studied. In Chapter 3 it is established the mathematical model developed, from the objective functions to the constraints used. Chapter 4 lists the electrical system data used, as well as all the assumptions taken to solve the problem. All the results of the simulations are presented and discussed in Chapter 5. Finally, Chapter 6 highlights the main conclusions of this work, the future works that can be done as well as the contributions of this dissertation.

Chapter 2

Blockchain

This chapter introduces blockchain technology, discusses its history, first applications and advances that led to the creation of cryptocurrencies such as Bitcoin or Ethereum as the most important blockchains that led to the revolution of this sector. Following, the architecture and features are explained in order to discuss why the Ethereum network is thought to be a better approach for the development of this project as it allows the creation of smart contracts, that will be explained further on. A section is also reserved for IoT and AI revealing their importance and what makes them essential technologies for the present or a possible helper for further development.

2.1 History and Present Interest

Developed in the early 1990's by Stuart Haber and W. Scott Stornetta, blockchain was an attempt to achieve a trustworthy way of storing transactions by using a timestamp to prevent the change of information [43]. This way, if data were altered the saved timestamp would be updated and the user would be able to easily recognise a change to that the file which was modified. A transaction is recorded in a list that is itself registered into a block. At first, each block could only store one transaction, what was not an ideal organization nor efficient, but by the incorporation of the Merkle tree the project was made more efficient by enabling the store of multiple transactions into a single block [44].

This technology went unused and its patent lapsed in 2004, four years before the arrival of the Bitcoin, first public usage of the blockchain. In the same year, Hall Finey introduced a system called reused proof of work (RPoW) that would prove to be as an important step for the creation of cryptocurrencies. This system used a non-exchangeable and non-fungible hashcash based Proof of Work (PoW) token and in return created an RSA-signed token that could then be transferred from person to person solving the issue of possible double spending of tokens by registering the ownership of tokens on a trustworthy server were all users were allowed to verify its correctness over time [45].

By the end of 2008, Satoshi Nakamoto, a pseudonym saw the possibility of combining both previous technologies and released Bitcoin's white paper [46], [47]. The first appearance of

blockchain in the real world, blockchain 1.0, may be considered as the era of the cryptocurrencies. Bitcoin is based on a hash POW algorithm but rather than using a single computer like RPOW, the double spending protection is provided by a decentralized peer-to-peer protocol for tracking and verifying transactions. Bitcoins, the protocol cryptocurrency, are used as a reward for miners who can earn them by creating new blocks which are introduced into the network and then verified by all the peers. Satoshi Nakamoto was rewarded with 50 Bitcoins for mining the first block from which he transferred 10 to Hall Finney, making the first Bitcoin transaction in 2009. Nowadays Bitcoin has the largest market capitalization, reaching its peak by the beginning of 2019 where its value was almost 20 thousand euros per Bitcoin. As of the end of June 2020, the market capitalisation is 148 590 000 000 and the price per Bitcoin is just over 8000 [48].

The POW algorithm is a major reason why Bitcoin is such a secure software. According to this procedure every node of the network has the same importance when to reach consensus, with consensus being the agreement of the network on the validity or not of a certain transaction [49]. This means that before a node can validate a transaction it has to possess a copy of the blockchain to verify that the data on the transfer logs are the true values, for example the number of tokens that each actor of the transaction owns. The process to create a new block with a PoW algorithm is showing in Figure 2.1. This process makes it much more difficult for malicious actors to corrupt the network as the network grows, since in order to reach consensus the majority has a final say, an identity would have to possess an enormous computing power to validate a false transaction what makes it extremely energy consuming. In fact, this tells us that the Bitcoin can suffer an attack but, in order to be successful, someone who wanted to make a network penetration would need more than 50% of total computing power to ensure its network version of the network was accepted as true and consensus could be reached, leading to a accomplishment of its malicious action. It's estimated that the cost of such an attack would cost more than 6 billion euros [2]. Although this protocol has protected Bitcoin from such successful attacks, there have been cases of stolen Bitcoins but this relates to the use by user of a wallet or cryptocurrency exchanges.

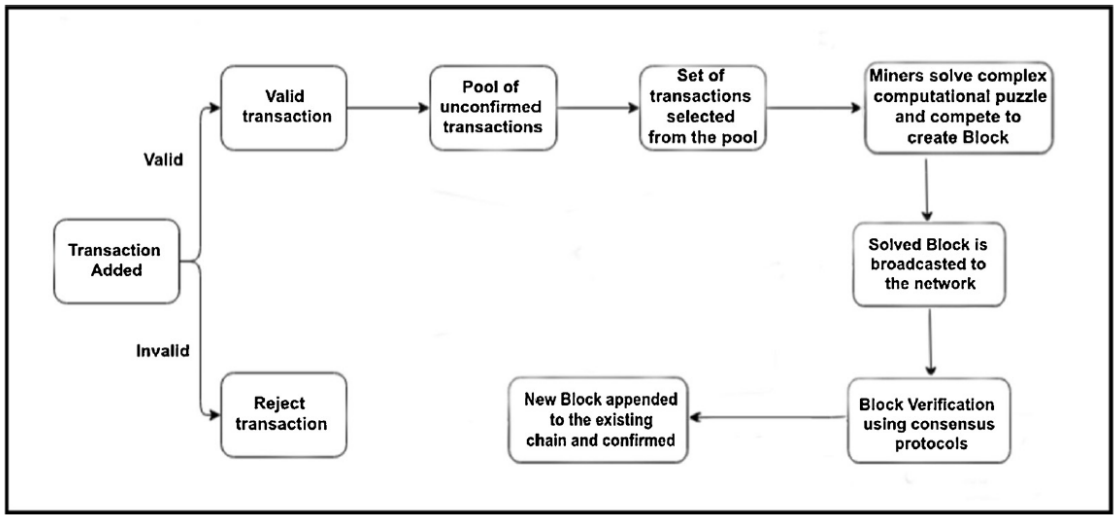


Figure 2.1: Block creation on PoW consensus protocol [1].

In 2013, Vitalik Buterin, a 19-year old programmer and co-founder of the Bitcoin Magazine, failed to convince the blockchain community that Bitcoin needed a scripting language for building decentralized applications (DApps) and started developing, on his own, a new blockchain-based distributed computing platform, Ethereum [50]. This new blockchain deferred from Bitcoin by allowing developers to construct smart contracts which consist on pieces of code written in a specific language, Solidity, that gets executed when certain criteria are met [51]. Smart contracts are compiled into bytecode that can be read and executed inside the Ethereum network. The Ethereum cryptocurrency is called ether and it can be transferred between accounts and used to pay fees for computation work needed to running transactions when executing smart contracts. Ether has the second market capitalization peaking also in January 2019 with a price of over 1.300 per Ether [52].

Although smart contracts became a reality with blockchain 2.0 [43], their concept was for the first time designed in 1996 by Nick Szabo as a way to substitute ordinary contracts that sometimes had enormous costs. They may be considered as the first step to reduce the need of an intermediary within transactions and they have low transaction fees compared to those traditional contracts implemented by regulatory authorities. Since the verification of transactions is now completely executed by machine, the need of human intervention is greatly reduced, making it a much faster process. However, the term smart contract has raised some arguments relating to its name. In fact, they are not smart at all. They have no processing power, have no ability to learn, just to check the relevant criteria and perform actions which are autonomous processes. Second, they may not even be considered legally binding contracts. A contract has some elements that must be considered that will not be present in smart contracts in most cases. A major foundation of a contract is that the person agreeing to the contract should be of a certain, legally accepted age but, in the case of smart contracts, there is no way of knowing the age of an peer since the blockchain offers anonymity or pseudo-anonymity what may result in an illegal agreement [53]. Another aspect to be solved is the case when the terms agreed in the contract are not accomplishment of and how to allocate consequences among the contractual parties. These are some problems that need to be solved if this technology is going to be part of our society.

As blockchain is continuously evolving as many versions continue to appear in the market the next step is its evolution is the implementation of Decentralized Autonomous Organizations (DAO), the so-called blockchain 3.0 [54]. DAOs are nothing more than a collection of smart contracts that operate without the need of human intervention [55]. Once initiated, full autonomous DAOs can never be stopped nor their business logic changed which can be a huge disadvantage. Traditional entities adapt over time to better suit their changing environments and these kind of DAOs won't be able to do so and, therefore, not be able to correct any programming errors that may occur. Semi-autonomous DAOs may be a solution to this particular problem, where the institution is able to operate by itself and execute all transactions with its computational power but an authority was able to modify its foundation when errors were found or some functionalities needed to be add in order to keep the progress. Otherwise, any time any further development was tried to be released, users needed to register to another institution.

Nowadays, blockchain is of high interest all around the world due to its security and reliability. According to hacker Noon, just in 2018, there were 24 conferences focused on discussing this technology, where 10 of those were primarily directed on the application to the energy sector [27]. As powerful economies like China, Russia or the United States are competing to become a leading developer in this area the European Union is also making maximum effort to stay in the front line of research. As for the year 2019, the EU granted 80 million euros to projects that would apply blockchain to many diverse sectors, and the EU is planning to keep increasing this investment. Besides the EU, also their state members have their investments in blockchain. Countries like Germany, Austria, Spain, Lithuania or France are heavily investing in this technology or regulating new ways to adapt it.

Blockchain's tokens are also being used by startups to increase founding by Initial Coin Offering [56]. These ICOs are in some cases fraudulent schemes where the promised projects are never intended to be finished and are just used as a way to raise money for the company. This may lead to a decrease in investment since no investor wants to have their money lost over some non-viable solutions. This requires countries to clarify the legal and regulatory framework, primarily resolving tensions between individual's data protection as well as legal, fiscal and accounting status of tokens clarification with rules surrounding the exchange of value [57]. Although many new companies are emerging a lot of existing corporations are also working with new technologies, for example, Microsoft, Apple, Intel, IBM and much more. Close to 60% of large corporations are thought to be in work or consideration of start working with it. Even financial institutions do not want to be left behind. Looking into blockchain as a possible threat for their future or a potential asset for security, they are increasingly entering this outside market, with approximately 90% of worldwide banks working in exploring its use in electronic payments [57].

2.2 Basic Principles and Architecture

The main characteristic of a blockchain relies on its visibility to all the participants. A blockchain can be classified as a public, private or a consortium network [2], [46]. In the case of Bitcoin or Ethereum, which are public blockchains, any user is allowed to create a personal address and start interacting with the network by submitting transactions, hence, adding entries to the ledger, or access transactions history and information on the database. In other situations, such as IBM's Hyperledger, JPMorgan's Quorum or R3's Corda, blockchains are private. Being private means that no one can be part of the network without invitation and only some chosen entities have permission to access private user data. Lastly, consortium blockchains may be considered as a combination of the last two, composed of specific groups with permission to edit the ledger. Public and private blockchains can also be referred to as permissionless and permissioned networks, respectively, and choosing to implement one or another should differ on the purpose it is being designed for. As any of those comes with advantages and disadvantages they must be considered when starting a new project.

Permissionless blockchains, as appear on the left side of the Figure 2.2, need to be decentralized, which means that no peer has a more important role inside the network than others as all peers function by the same rules and have the same rights [46], [1]. This implies that no entity has the authority to edit the ledger, shut down the network or even change protocols. These kinds of chains are based on consensus protocols, which means the agreement has to be reached for a transaction to be executed, most importantly the user log registries. Reaching consensus can also depend on the chosen algorithm but, as an example, for consensus to be reached more than 50% of network's peers should validate log data to allow the transaction achievement, otherwise, it's cancelled. Consensus algorithms could not be possible if transparency was not an essential characteristic of this network. All the participants must be given full access to all information, from addresses to transaction processing and transaction history. One last aspect of these networks if the presence of a financial system. While validating transactions, peers act as miners, that in case of Bitcoin help to solve complex mathematical equations, as get rewarded with small fees or blocks for executed work. These fees come as an incentive for new users and are expressed in tokens that can grow or decrease in value on the exchange market creating a group of self-interested individuals that contribute towards making the system more trustworthy.

Permissioned blockchains, as shown the right side of the Figure 2.2, are closed ecosystems preferred by centralized organizations that utilise the network for internal business operations, choosing the ability of an entity to be part of the network and approving which nodes can run the consensus algorithm giving them access to recorded information [46], [1]. As there is a central power, this kind of networks does not act as a decentralized network since it does not require the need for transparency as permissionless blockchains do. These networks represent a governance model based technology rather than a consensus-based one. As the central power has the right to decide who can be part of the consensus algorithm the fact of choosing the nodes from which power will be distributed than the algorithm to reach validation becomes more important. Having fewer nodes verifying transactions improves processing speed as it keeps networks safer against outside attacks by better protecting confidential data [3].

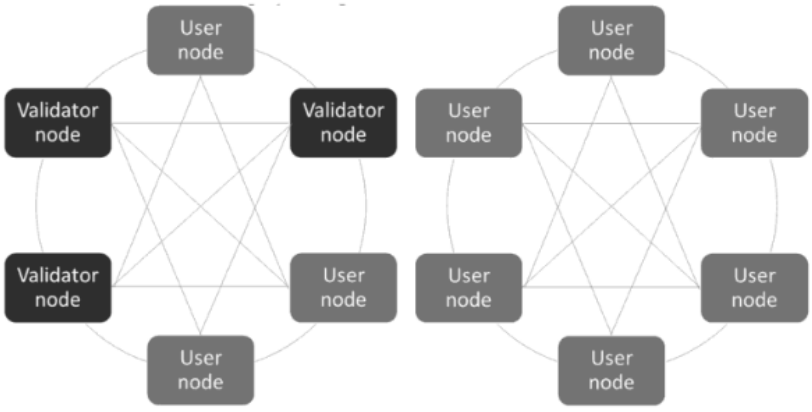


Figure 2.2: Difference between permissioned (left) and permissionless (right) blockchains [2].

A central topic when talking about blockchain is security. It's known that to date no successful malicious attacks have been performed on Bitcoin or Ethereum blockchains. There were some cases of stolen currencies but that is a matter of wallet security more than network security [2]. Blockchains are very fault-tolerant primarily due to their strong consensus algorithms, this means that when a node tries to falsify a transaction or information already stored in the database other nodes do not let those operations occur because they have access to all the information stored and recognize it as false. The best way to keep the trustworthiness and security in the network is then by making sure that the latest version of data is delivered to every node in the chain. In this case, all the involved parties in the transaction validation operation are comparing the transaction logs with the most recent ones.

Even though Bitcoin or Ethereum are the most common referred cases to illustrate the function of blockchains, a cryptocurrency is not the only application of this technology. Since the blockchain does a perfect job storing information in their ledger, the store of already accomplished transactions from payments or asset registration would facilitate data regulation and organization [3]. This characteristic makes the blockchain an exciting match for co-working together with AI [27]. Even the best AI in the world would be useless without information and because the blockchain is able to store almost an infinite quantity of it, if an AI had access to that information it would lead to much more faithful results.

However, the fact that blockchain can store infinite quantities of information with the capability of constant growth, in an appended-only database, to which information could only be added and not removed, creates a sense of mistrust from some potential users [58]. No one wants to see their personal information shared with the outside public and much less that their information is kept stored in the eventuality of their exit. The truth is that data can be erased, but with the consumption of a large amount of processing power, that needs a lot of energy to be executed resulting in high costs. The reason why they are in most cases not considered. But the DGpr (General Data Protection Regulation), gives individuals the right to amend their data, to remain accurate or with certain exceptions erased when no longer needed [57]. How blockchain will incorporate the needs of GDpr or other data privacy regulations is a crucial research question to answer in the coming years.

2.3 Features of the Technology

In this section, two key features of the blockchain technology are described. These key features are the data encryption ability which guarantees privacy of any data transmitted and transactions within the blockchain which allow peers to interact directly without the need of an intermediary third party.

The data encryption function of blockchain underpins the immutable nature of blockchain, that is the blocks cannot be altered once accepted [59]. The encryption of a blockchain system is done through the use of hash functions. A hash function is defined as a one way, or non-invertible encryption function which maps a set of inputs directly to a set of outputs. It is a deterministic

function in the sense that the same collection of inputs will always result in the same outputs. This is the key aspect of the blockchain as it is constantly updated, any alteration to a previous transaction will alter the has function of the block and be instantly recognisable as being modified. Even very small changes in inputs of a hash function result in drastically different outputs. New blocks within a blockchain should always contain the has of the previous block as this is the link to ensuring that there can only be one accepted version of past transactions.

In order for transactions to function within the blockchain system, security of the transactions is key. This security is guaranteed through the use of a key pairing system [60]. Each user within the system has a public and private key. These keys are related mathematically through the use of an asymmetric elliptical curve function [49]. Only the user knows their private key while they share the public key when making a transaction. Due to the property of intractability of elliptic curve cryptography, the private key is nearly impossible to derive from having only the public key. As illustrated in the Figure 2.3 the sender initiates a transactions by creating a signature for the transaction. This signature comprises of her private key, the receivers public key and the data of the transaction. Then the sender's public key, which is known, is used to verify that the sender did originate the transaction and this is verified by the miners.

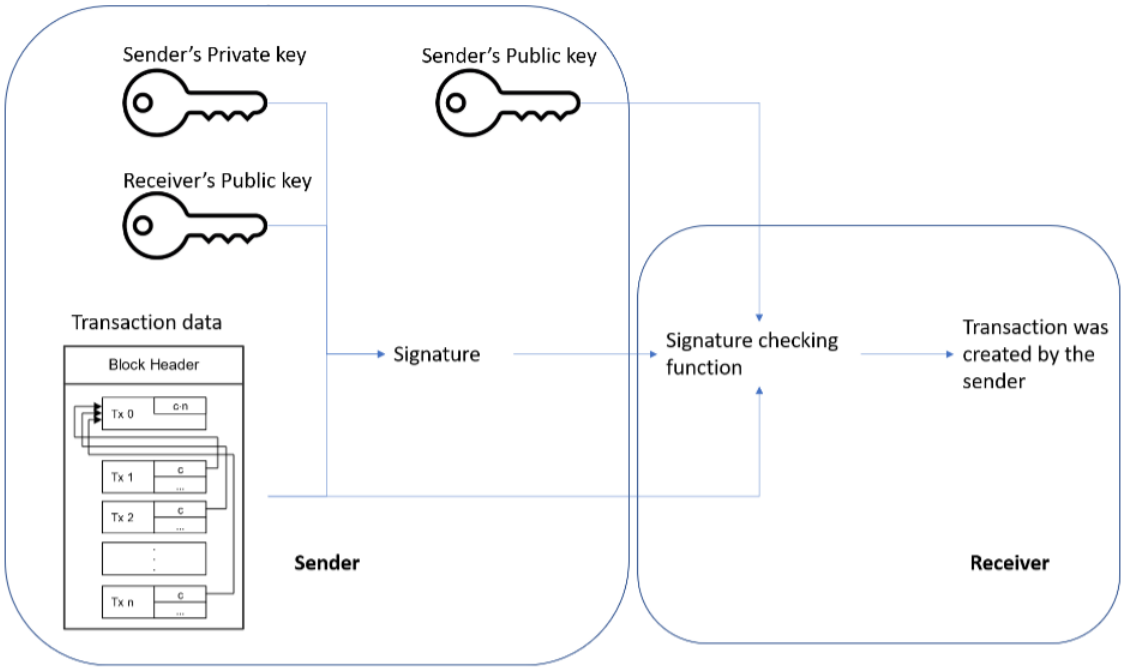


Figure 2.3: Process to create a transaction in blockchain

2.4 Gas

Like any proof-of-work P2P system, Ethereum is heavily dependent on the hash rate of their miners [49]. More the miners, the more the hash rate, more secure and fast the system. Ethereum has smart contracts to assure the follow of steps to a final purpose. Miners act as entities who execute the smart contracts and assure those steps are followed and the final purpose is reached. This is a process that benefits both parties: the network is more secure being executed by the miners and the miners receive an incentive for doing this job as is presented in Figure 2.4.

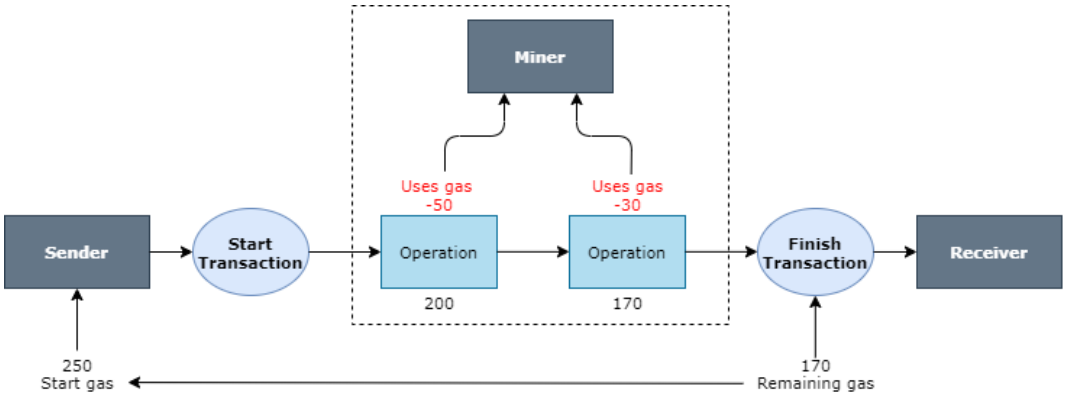


Figure 2.4: Gas paid as fee

As previously explained, computational power, in Ethereum, is measured as gas that must be paid as a fee for some entity run a function, the miners [61]. This gas amount is in less quantity when compared to energy price but it has to be considered. Some functionalities have no gas consumption such as the reading of data from the database, but others do like adding new information or altering it [62]. Mathematical operations also consume gas and also the more complex the more gas needed. Every transaction requires at least 21,000 gas according to the Table 2.1.

Similarly, when a transaction is made through blockchain, miners help pack the transaction and put it in the new block to complete. During this process, miners consume computing resources and thus the sender must pay a gas fee to them to compensate for the resources consumed. The more complex the transaction is, the more computational resources are consumed, and thus the more gas is consumed. Different transactions will generate different gas costs [62]. Miners will stop executing the operation when gas is run out. The sender can get the unused gas back and do not have to worry about setting an exceedingly high level for gas limit because if there is any unused gas, it will be returned immediately to the sender's account.

The sender always pays the miner fee even if the transaction is failed. If the amount for the gas limit the sender indicates is too low, then the transaction is considered invalid and will be rejected with an "Out of Gas" error, and the gas spent for computation will not be returned to the sender. It is important to understand that the sender always pays the miners for the computation, regardless

Table 2.1: Gas price chart

Name	Value	Description
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{eztcode}$	700	Amount of gas to pay for operations of the set $W_{eztcode}$.
$G_{balance}$	400	Amount of gas to pay for BALANCE operations.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for a SSTORE operation when a storage value is set to non--zero from zero.
G_{sreset}	5000	Paid for a SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{suicide}$	24000	Refund given (added into refund counter) for suiciding an account.
$G_{suicide}$	5000	Amount of gas to pay for SUICIDE operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a call operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for a called contract from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SUICIDE operation which creates an account.
G_{ezp}	10	Partial payment for an EXP operation.
$G_{ezpphyte}$	10	Partial payment when multiplied by [log_{250}(exponent)] for the EXP operation.
G_{memory}	3	Paid for every additional word expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the *Homestead transition.*
$G_{tzdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{tzdatanonzero}$	68	Paid for non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logtopic}$	8	Paid for each byte in a LOG operation's data.
$G_{codedeposit}$	375	Paid for each topic of LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (round up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for COPY operations, multiplied by Words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.

of whether the transaction goes through or not. This payment is always done in ether, even though the token being used in the project is another ERC20 standard [61].

Following there are some terms clarification, some which appeared above:

- Gas:

  Gas is a unit of measurement to calculate the amount of computation needed for a particular operation.

- Gas Fee:

  Gas Fee represents the miner fee for a particular action or transaction. Gas Fee is composed of two parts: Gas Limit and Gas Price.

- Gas Limit:

  The Gas Limit refers to the maximum number of Gas a user is willing to pay for performing an action or confirming a transaction (a minimum of 21,000).

- Gas Price:

  The price of Gas is the amount of Gwei that the user is willing to spend on each unit of Gas. WEI is equivalent to ETH in the way that a cent is equivalent to the euro, it is the smallest denomination of the currency.

The higher the gas price the sender is willing to pay, the quicker that the miners will confirm your transaction since the reward of the miners is higher. As some would expect, the higher the proposed gas price, the higher the chances that the transaction will be included in the next block since this is what incentives miners. The default gas price on most interfaces is 20 GWEI, which should be sufficient to get a transaction in the next couple of minutes. Increasing the price to 40 GWEI will likely get you in the next block.

2.5 Prospects in the Electric Market

The blockchain is yet a recent technology with few applications in use around the world, but it is thought that the electric market could be the most advanced one to include this new way of thinking, revolutionizing all its structure and way of working. A huge uncertainty is, however, a reality, due to a so big change the blockchain could bring.

The most obvious answer is the use as a way to connect two parties in a peer-to-peer architecture, making the process of energy transaction a lot more efficient and giving it the ability to trace it at every stage of the transaction [59]. It would act as a platform where buyers could place their offers of energy over a certain quantity of value and where consumers would analyse the presented offers and chose the ones that better served their needs. As producers would be rewarded by their contribution, this is, by receiving money for the energy they're supplying for someone else, this

would be a big incentive for them and also for consumers that would see energy prices reduced in a market where the supply versus demand is more equalized than the relation presented in the current market system [63].

It was already stated that blockchain works as an excellent database [27]. When introduced in the electric market, blockchain could store consumption data turning a consumer more aware of their impact and energy consumption's costs. This could be an open eye and result in a reduction in energy consumption. In the year of 2016, there were invested 47 billion euros in digital upgrades to electric power system, adding sensors, modernizing outdated information and communication technology platforms and more [3]. This reinforces the idea of an almost-ready system for the embarrassment of blockchain.

Data storage could discard a huge problem this electric market arrangement always faced and tried to minimize over time, that is the uncertainty of energy consumption and forecast of energy production. The system, that would have real-time information about the operation of the power grid, could become composed by more flexible power generators to meet the needs of inflexible and unpredictable consumers [3]. Typically, customers have had to inform utilities that a blackout has occurred what now would be of easy perception.

Energy transactions would also become automated, putting aside human work that sometimes occur in problems, for example asymmetric information between regulator and regulated entity [59]. Last but not least, a peer-to-peer energy market system could turn the present market into a more environmental friendly one since prosumers would turn to renewables like solar panels for their own energy production.

Some entities have already pronounced about this technological breakthrough, like Greentech Media which considers blockchain and smart contracts with great use when applied to Renewable Energy Credits (RECs): "For RECs, blockchain can replace legacy platforms and reduce the possibility of double-counting. Developers can design these REC trading platforms to conform to existing regulations [27]." RECs are an important piece to recognize ownership of renewable generation and also their role of account and track the energy could be of a better organization.

2.5.1 Opportunities

There are some applications that could take the most value from blockchain use when applied to the electric market. The following list and showing in Figure 2.5 reference the ones with most importance, some already being in development and as an attempt to reach total integration others reveal pretended future work [3]:

- Peer-to-peer transactions:

  This architecture takes the most important place in this research, being the idea tried to be investigated and placed in action. A P2P network happens to be the most intuitive and popular among the presented below when applied to the electric sector.

  The purpose of this idea is for costumers to trade electricity to other costumers, for example the excessive renewable production the have in their own home, that could come form

solar panels or a small-scale wind turbine [64]. This architecture reveals to be a purely decentralized system as there is no intermediary in a transaction between to peers in the network.

However, a truly decentralized system can be thought as a utopia. The electric market is considered to be a highly regulated rector and the drastic change is very difficult to become a reality, and even more in third-world countries, despite the good work of some startups.

These networks don't change the actual flow of energy, only act as possibility for virtual transactions, and for this reason it continues to rely heavily on today's transport and distribution grid that, despite of outdated, are very difficult to replicate in a true peer-to-peer system.

- Grid transaction:

  These networks are more likely to become a reality and gain more commercial importance and support from regulatory authorities. The main reason is because they don't become completely put aside from the system with this topology so it is of their interest that this architecture is chosen from the list presented [65].

  Although an intermediary continues to operate in this system, they do not support as much power as they do in the present market architecture so the bulk of prices they now practice, to overcome costumer's demand or production and distribution losses, is not likely to continue happening anymore. These distributed energy resources can defer the needs of infrastructure upgrades, which have been done for some time to keep the system up to new innovations, and even help stabilizing important grid parameters such as frequency and voltage.

- Energy financing:

  As it was referred before, blockchain can be developed as a way to raise funds in form of ICO. These ICOs can help in renewable energy project's development and extend the pool to potential investors and enabling a multitude of smaller investors to supply capital [28].

- Sustainability attribution:

  Current systems are centrally managed and willing to faults and errors. The most immediate application of blockchain technology is the record of history actions, primarily consumption data. A decentralized blockchain network could enable transparent, accurate, and friction-less tracking and trading of these attributes, which would accelerate clean energy deployment and carbon emissions reduction [66].

- Electric vehicles:

  As electric vehicles become more present in today's society, other uses must be considered for their use and not just travel. Electric vehicle's batteries could be used as a way to source the grid in times of most needs in case its not needed and then charged in low peak hours [28].

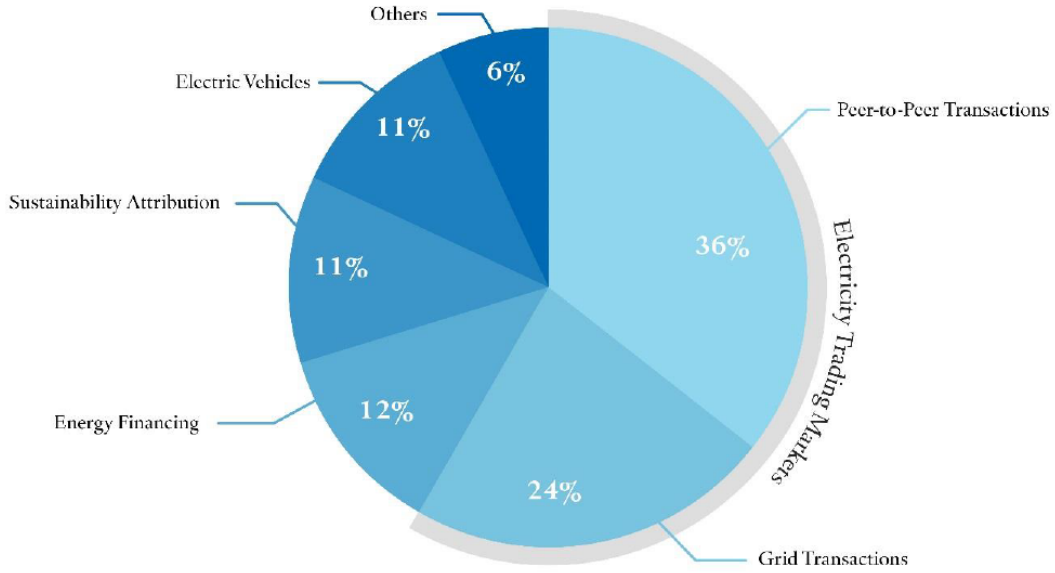


Figure 2.5: Use cases of blockchain on the energy sector [3].

2.5.2 Challenges

The first challenge blockchain faces is the ability of do a high number of transactions per second. In today's version Ethereum is only capable of making a few dozens of transactions per second which is nothing when compared with the thousands that visa is able to do. Of course this will improve over time. Visa also started by only being able to do a few transactions per second so developers must continue working to improve it's processing velocity.

When thinking about blockchain appliance to the electric market, the first idea is the replacement of today's centralized market power by a decentralized energy trading scheme. However, it will most likely not occur knowing that this is a so highly regulated market [3]. Whether or not total eradication of third parties is desirable for a perfect market system is yet open to discussion since in many cases they perform important functions and become difficult to substitute. Instead of full decentralization, blockchain could make them re-think the role of the regulator entity in many industries and contexts, forcing a re-pricing of their services [57].

The biggest problem to blockchain integration is that power utilities are risk-averse entities, slow to overcome changes in the system, in part because they face security from regulators as well as pressure from shareholders that seek stable returns [3]. The best response regulators could have right now is being adaptable to a changing environment and take to account the telecommunication system deregulation, and the fact their regulators where not able to predict how changing technologies and economics would shape the industry. [27]

Policymakers will also have an important role to play if the aim in to integrate blockchain into this regulated system. Investment into understanding the technology and how they could interact is a primarily point of choice [3]. But also some works should be put in practice to permit experimentation, creating small-scale demonstrations because, only this way its function can be

of total understanding and help to chose the best approach based. Choosing the most ready one of the already presented ideas does not mean it will be succeeded. The new system must be well embraced by costumers and sometimes not only one must be chosen but a combination of some could lead to a optimum result.

2.6 Chapter summary

This chapter described the blockchain technology. First, the origins and steps that lead to the foundation of the Bitcoin and then Ethereum, the blockchain in which this project will the developed on, were discussed. Various applications of this technology to the electric system were explained, these advantages are largely due to its decentralized and secure network. Smart contracts were also presented and it was shown that they originated prior to the development of the blockchain and now are a major aspect of the blockchain technology with a promise to automate actions which otherwise would be costly due to human involvement.

In the following chapters the project will be presented. The organization of the work will be detailed and a deep description of the methodology followed, especially highlighting the steps that lead to the finalization of the system.

Chapter 3

Software Development

In this chapter, the design of the blockchain based trading system is discussed, including the various steps that were followed to ensure its successful completion. The barriers and issues identified as well as solutions created during the development will also be discussed. Various functionalities of the developed code will be presented.

This chapter begins by introducing the chosen Integrated Development Environment (IDE) and then the various entities within the market environment are presented. The market design and contract structure are then discussed in depth. Most importantly the energy transfer method between peers always ensuring a secure payment is shown. Gas usage is also taking to account since the more gas is consumed the higher the fees the users must pay to accomplish any function which may increase costs for them.

3.1 Remix

The choice of a code editor for the development of the software is crucial. There were a lot of applications to chose from but in the end, it was clear that the one that best suited the requirements the web browser-based Remix, for the features only it could provide that will be explained shortly [67]. The version of Remix used was v0.10.1 and continues to release constant updates since it is still under a significant development phase. The IDE automatically updates to the newest release so the developers are always using the latest version.

Some features are common to other IDE's, such as code developing features: syntax highlighting, auto-compling, auto-saving, and some more which help with the writing and easy understanding of the work done. But the major advantage comes with specific characteristics for solidity language. The Remix IDE provides many environments to build, execute, and compile the contracts. The environment used for this project was Javascript VM, a sandbox blockchain implemented with JavaScript in the browser to emulate a real blockchain. Otherwise, an injected web3 environment can also be used, where a provider injects an object such as Metamask, connecting the user to the blockchain [68]. Metamask acts as an account manager, storing user's keys while isolating them from the project context. For the developer side, they only interact with globally

available Ethereum application program interfaces (APIs) and whenever a transaction is requested, Metamask will prompt the user to, grant more security for the user's wallet access. These are actions that the IDE does automatically for the user when using other IDE's the user would have to write the code that let to the blockchain connection what helps to save time in testing fazes.

The menu with directions for compiling, run, settings, debugger, analysis and support is easily find on the left side of the window as shown in Figure 3.1. When compiling any contract, the user is interacting with the simulated blockchain network provided by the IDE. The user may execute transactions to prove the validation of certain actions. When running, the user can also select values for gas limit and gas price and see values stored on the contract as can also be seen in Figure 3.1. It also gives the user a list of errors or warnings when the best practices aren't being followed in the left side menu. A common example is the gas usage. Remix tends to recognize when users make a dangerous action that may consume a high value of gas and tell him to try to reach a better approach. This is an important feature of this IDE as it helps the developer to write efficient pieces of code with the minimum computing power needed for its execution. A terminal is also integrated into the App as showing on the bottom of the image.

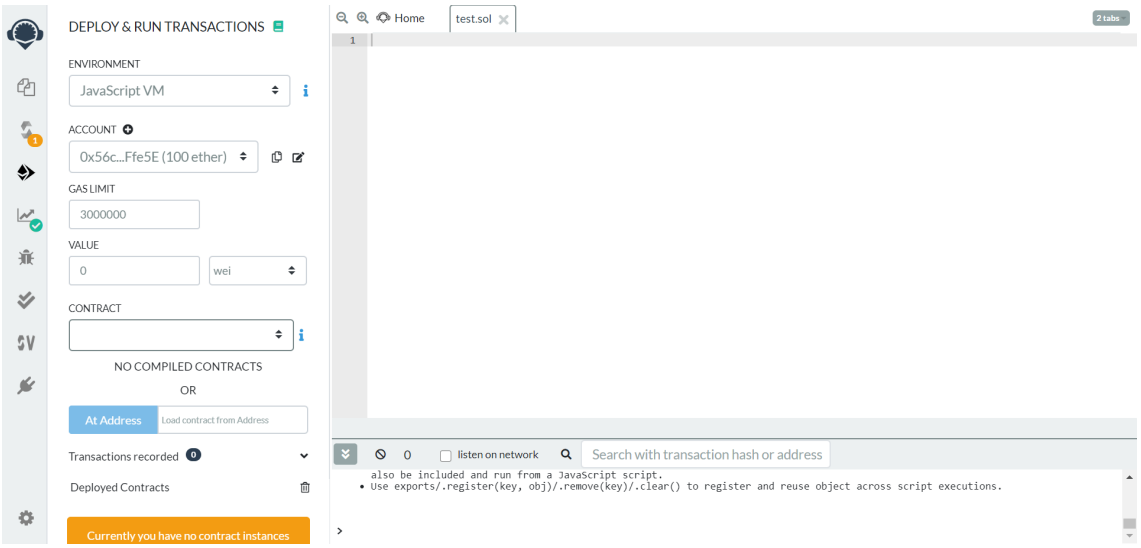


Figure 3.1: Remix IDE

Even though, for the development of this project, there was no need for a manual connection to the blockchain network, or a manual creation of a simulated one, when learning the Solidity language these actions were also applied to better understand the advanced features of the Ethereum blockchain. Initially, the Sublime editor was used which meant that code was manually compiled and contracts were manually deployed [69]. To reach a machine readable version of the code able to be understood by the Virtual Machine the code shown in Figure 3.2. This code has to be changed when compiling different contracts as had to be run every time a change was made in the contract. This is time consuming to do so a more automatic way of doing this was identified.

```
const path = require('path');
const solc = require('solc');
const fs = require('fs-extra');

//delete entire build folder
const buildPath = path.resolve(__dirname, 'build');
fs.removeSync(buildPath);

//read Campaign.sol from contracts folder
const campaignPath = path.resolve(__dirname,'contracts','Campaign.sol');
const source = fs.readFileSync(campaignPath,'utf8');

//compile both contracts with solidity compiler
const output = solc.compile(source,1).contracts;

//make build path
fs.ensureDirSync(buildPath); //looks to see if directory exists, if not creates it

console.log(output);
for (let contract in output) {
    fs.outputJsonSync( //writes JSON file
        path.resolve(buildPath,contract.replace(/:/g,'')+'.json'),
        output[contract]
    );
}
```

Figure 3.2: Manually compiled code

Figure 3.3 shows the steps needed to reach the connection to the simulation network provided by a web3 object. This code was mandatory when any contract was to be deployed as there as a need to have a connected wallet to the project. The wallet was provided by Metamask which provided access to the main Ethereum network and to several test networks and connected to the project via the Truffle provider [70]. Lastly, every command had to be executed on the laptop terminal. To run the codes for compiling, deploying, or testing the contracts. These are the steps that Remix helps the developers by doing it automatically turning it to an excellent time-saving application and was a major reason for selecting this IDE.

```
const HDWalletProvider = require('truffle-hdwallet-provider');
const Web3 = require('web3');
const compiledFactory = require('./build/CampaignFactory.json');

const provider = new HDWalletProvider(
    'blast glad base gown load pear hunt omit message chicken loop dry',
    'https://rinkeby.infura.io/v3/ccd2c81e2be34010a8901ddf487b5f3f'
);

const web3 = new Web3(provider);

const deploy = async () => {
    const accounts = await web3.eth.getAccounts();

    console.log('Attempting to deploy from account', accounts[0]);

    const result = await new web3.eth.Contract(JSON.parse(
        compiledFactory.interface)
    )
        .deploy({ data: compiledFactory.bytecode })
        .send({ gas: '1000000', from : accounts[0]});

    console.log('contract deploy to', result.options.address);
};

deploy();
```

Figure 3.3: Manually deployed code

3.2 Entities

As discussed previously this is a user-oriented project, with the consumer in the center of the development of the system. Therefore, there are some present entities whose characteristics must be highlighted to better understand why they are designed this way. Each entity has different rights, mostly depending on their ability to execute the different functionalities in the code.

Understanding the chosen entities was the first step for the development of the project. To better explain their abilities, there is a use case diagram in Figure 3.4 showing relations between the different entities and functionalities they might have.

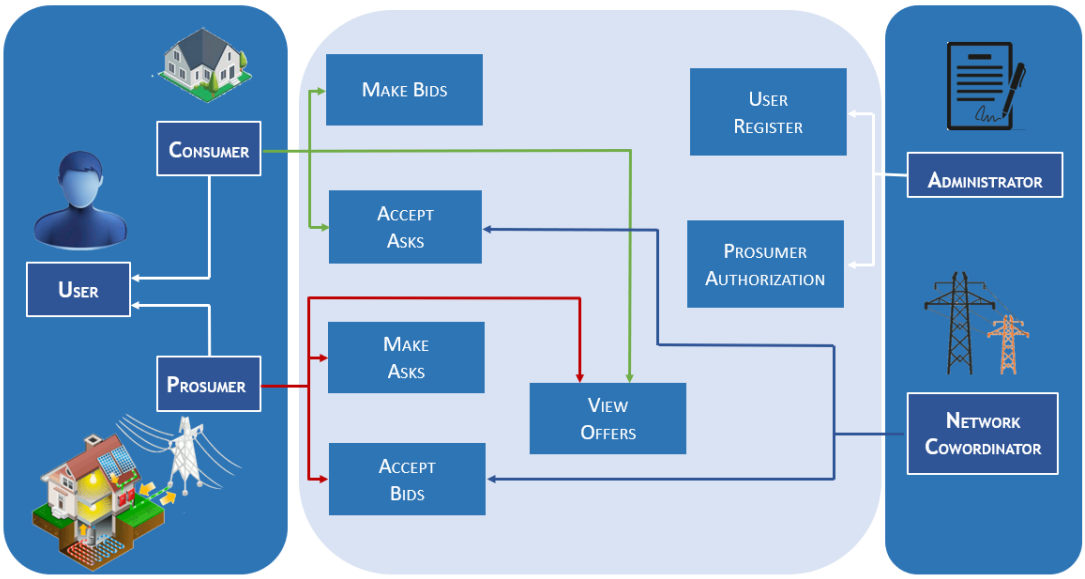


Figure 3.4: System's use case

3.2.1 Administrator

The administrator is a central entity without whom the system could not function properly especially concerning protecting the user's data. The administrator is the only entity responsible for adding or removing a user from the system as well as giving authorization for users to sell their energy, turning them into prosumers within the system.

The presence of the administrator provides higher safety and security for the system not taking its decentralized focus of development to what it was designed for. This entity does not affect the market functionality but only keeps users information secure by not allowing any other entity to access the information apart from completing a transaction. A choice was made to keep the administrator separate from the functioning of the market so as to limit their incentive to manipulate the market.

3.2.2 Consumer

When a user is registered in the system by the administrator, they are automatically given a consumer allowance. The consumer is then able to buy energy from the market and to make demands for it. Consume profiles may vary from time to time and even for different days so, the user needs to take this into account to understand the amount of energy they may need to fulfill their needs. This responsibility, which is for now assumed his own, may in the future be given to a digital asset, such as a Home Energy Management System, which calculates the user needs, automatically informs the consumer and then suggests the required bids to meet both the energy demand and comfort of the consumer.

3.2.3 Producer

A producer is a user to whom the ability to sell energy was granted by the administrator. A producer is considered a prosumer in the system since he can execute demands or offers of energy as well as accept them. In the real world, a prosumer would be a common user with access to some sort of small scale distributed energy resource, most commonly a renewable source of power generation. These generators could be some photo-voltaic solar panels or some small scale wind turbines for domestic use. Electric Vehicles (EV) could also be taken into consideration since they can provide energy as well as receive it. The flow of energy in an EV is bidirectional and then users may charge it or discharge it as they need.

3.2.4 Network Coordinator

The Network Coordinator (NC) operates in the same way as a prosumer, from the system's point of view, since the NC is authorized to buy or sell energy in the virtual market but is an external entity to the micro-grid. The NC can operate as the supplier of last resort both supplying any shortall in energy as well as buying any excess energy.

This entity distinguishes itself from the individual prosumer as it operates at a lower priority than they do, in other words, there is a clear priority for the market to utilize energy produced within the community first, and then only if needed rely on the NC. A consumer only buys energy from the NC if they could not buy energy to other prosumers and a prosumer only sells to the NC as a last resort. For simplifying purposes, the NC can not reject any demand, he is forced to buy or sell the energy in last resort.

The NC and administrator could be one entity with the combined responsibilities of both already detailed. However, for better understanding and division of tasks, they will be separated for the development of this project. The actual structure of the market and responsibilities of the NC and the administrator will depend on the existing regulatory regime.

3.2.5 Storing of agent's information

The inclusion of the administrator and the network coordinator are easy to do as there are only one of each within the system. The only relevant information from those entities are their private keys and so they are stored in an address variable in the developed system.

The difficulty increases when storing the user's information. For this project, we are going to work with a low number of consumers and prosumers but in a real-world application this number would be larger and so it must be taken into consideration. There is also another problem related to the time the network takes to execute a transaction. Ethereum can not execute many transactions per second and that can become a problem when it is intended to have a program with many users. This problem will not applicable to this project since the tests will be executed with a reduced number of users.

At the most basic level, for each person in the system, we would have to store three variables: the private key, the user, and the prosumer authorization. When it comes to a real-world application, we could add name, contact, and more characteristics of the user, which are not considered for now. This way we need a structure for a user with different types of variables. But since we have more than one user a dynamically sized type of variable is needed.

Initially an array was considered where each position points to a user's structure. This could be a solution but has been stated previously, code in Ethereum needs gas to run, and the more complex the code gets the more gas is consumed and so the more a user needs to pay a fee to execute the functionality. Using an array, each time we wanted to access a user data we would need to go through each position of the array, compare the user's private key of that position with the one we want, and if they are equal access the rest of the information. If we have to go through all the entries in the array is can be expensive. As seen in the Figure 3.5 the code has a private mapping named map_users where each position points to a struct of type user also presented. The struct has four variables. One to store the name of the user. The second represents the energy that each user owns. There is no way to measure this amount of energy so it is assumed each user has a hard coded value in this variable. The last two booleans represent the allowances given to the user by the administrator. The consumer boolean is considered in case a user causes any infraction and consequently is forbidden to use the market.

```
struct user {
    string name;
    uint energy;
    bool consumer;
    bool producer;
    }

mapping(address => user) private map_users;
```

Figure 3.5: User Storage

For this type of data organization, a better solution can be reached, a mapping, presented in Figure 3.6. A mapping is a kind of variable that uses a hash function to calculate the position of the information dependent on the key is given. Since no two keys have the same result from the hash function no two user's information can collide. In this case, the better approach for the mapping key is the user's blockchain account private key. This way, each time we want to access the user's data we only have to provide the key and the user's struct will be returned without having to go through the mapping. This saves a significant amount in gas consumption.

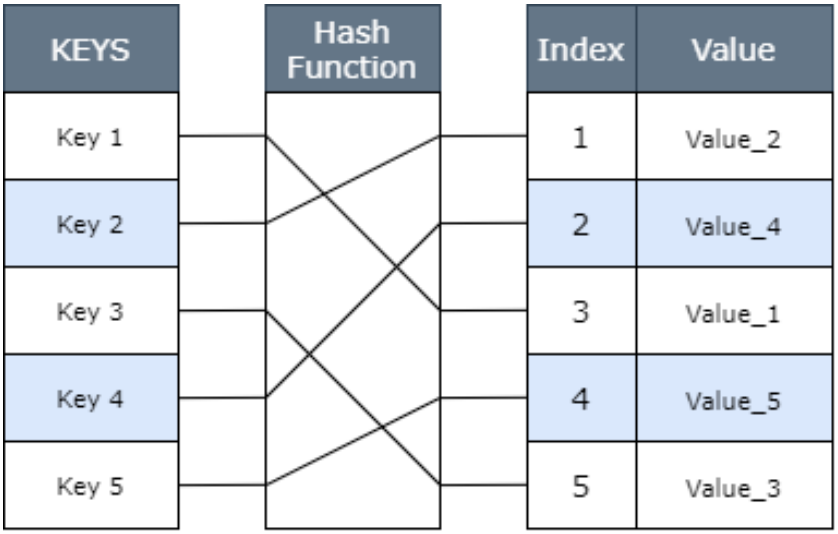


Figure 3.6: Example of a mapping

3.3 Demands and Offers

As the major purpose of the system is the exchange of energy between peers it was necessary to find an easily readable way to store the needs that each user has, the orders. According to their allowances, a user may execute submissions to the market for selling energy if they forecast an excess production during the next day or to buy energy, in order to meet their consumption. Energy demands and offers are referred to as as bids and asks, respectively, explained in the following paragraph.

- Bids:

  The energy demands are referred to as bids. They can be made by all the users, consumers and prosumers.

- Asks:

  The offers are referred to as bids and are only able to be made by prosumers.

The asks and bids are organized in 24-time blocks of one hour each, representing one whole day. The user may then take perceptions of their energy needs for each period as well as prosumers forecasting how much energy they may have to spare at different prices for different periods according to what they expect the demand to be for that period. Prosumers have a monetary interest in this market and so may try to increase gains as well as consumers shall be interested in reducing expenses. Each user has access to 24 different bids and asks, if a prosumer, for a single day and is only able to process one kind of order for each period i.e. a prosumer is not allowed to offer energy at a high price and then request energy at a lower price during the same time block.

The Network Coordinator is not allowed to make demands or offers of energy. As explained this entity only operates as the last resort and so, only sells or buys energy when the other peers do not fulfill everyone's needs.

Store of information For the storage of the bids and asks, there is something we may consider that was not needed until this point. When a consumer, for example, is trying to buy energy for a specific period, all the asks made on that block must be easily accessed without too much computing power, which otherwise could bring high gas costs and so, high fees. In the other hand, the system has to know which peer made that order so it can inform that specific peer they are about to reach an agreement and with whom.

Having this in mind we follow the same principle as in user's data as Figure 3.7 shows in a small excerpt of code. Since we have multiple types of variables to store in a single order a struct is used. The same type of struct (order) is used for both bids and asks but they are divided in the two matrices.

```
struct order{
    address agent;
    uint energy;
    uint price;
    bool active;
}

order[24][] private list_bid;
order[24][] private list_ask;
```

Figure 3.7: Storage of order information

We may look into each matrix as 24 different arrays of variable length. This method for storing the orders was chosen because when looking for an order to choose we may want to search through all of them, if needed, in a while or a for loop. Different arrays store orders for different time blocks and for example if a user wants to see the bids ordered for the period 12-13h he must look into the matrix list_bid and go through all the entrances in column 12 as represented in the Table 3.1.

Table 3.1: Example of storing offers

Time Slot	1	2	...	11	12	13	14	15	...	24
User 1	-	-	...	-	6Kw	8Kw	-	-	...	-
User 2	1Kw	2Kw	...	6Kw	-	10Kw	8Kw	8Kw	...	-
User 3	-	-	...	8Kw	-	15Kw	9Kw	9Kw	...	2Kw
User 4	-	2Kw	...	-	8Kw	9Kw	-	-	...	1Kw

3.4 Token

The main basic coin in the Ethereum network is the ether, as the name indicates. Otherwise many other coins exist in the network but the ether is core of all the tokens. To create a new token a client must pay a ERC-20 (Ethereum Request For Comments) contract, in ether, the software will send the money to the administrator and return the token ICO for him. These tokens are not independent as they reside in the Ethereum network and need support from the chain distributed computational abilities.

Before the ERC-20 standard, the platform had to write a completely new code for each wallet transfer to create a new token to the network. With a lot of new tokens being created it became essential to have a base code for all tokens so that they could interact easily between each other. The standard possesses six mandatory parameters besides three optional, but recommended, for example, the number of decimals the tokens supports. The protocol not always grants the needs to create a token for the purposed it is needed not to assure useful, valuable or functional coins, but acts a base code.

3.4.1 Project application

Since the purpose of this project is to develop a virtual market, there needs to be something to exchange for a certain amount of energy. The first applications of blockchains were for the creation of a virtual coin and today they are easily found in different environments. They are everywhere, such as Bitcoin and ether, the most valued virtual coins, which were presented on Chapter 2.

This kind of coin is highly volatile, and this is a main obstacle for the wide spread acceptance of those as a major method of payment around the world since there may be a huge drop in value that makes it hard for users to evaluate their assets. Recently, there has been some development in a new kind of virtual coins, the stable coin, that as the name indicates has a more stable value. More similar to the coins we see in circulation, as euro and dollar. An example of a stable coin created in an Ethereum environment is the tether, with similar value to the American dollar.

For this project it could be used the ether since for test purposes Remix grants some virtual simulated Ethereum accounts with 100 ether each. Each account represents a user for the system. Nevertheless, a new coin will need to be created to better simulate a real application with not so high fluctuations in value as described previously.

3.4.2 Payment safety

A major problem that may occur relating to tokens is the ease of their loss in a cyber attack considering that this is not a regulated system. Their tokens are stored in virtual wallets which are connected with their blockchain account. This is an issue not related to the development of this project, although all the necessary precautions have been made in the development of this project and the other safety concerns are related to the underlying structure of the blockchain system. The only security the project can assure is that payment made is to the right person for the right amount. The store of users money is their responsibility.

In a blockchain, not all functions may be accessed by everyone. There may be some functions the user cannot access and this depends on their type of access. To assure a secure method of payment for a transaction shown in Figure 3.8 function that could only be accessed internally was created as will be explained in the next section. This function was called when a user wanted to accept an offer from another function. This was the chosen way to make sure that the right amount of money was sent to the right person and from the right person, as well as the energy. No entity could call this function itself or alter the values from those for the agreement was made.

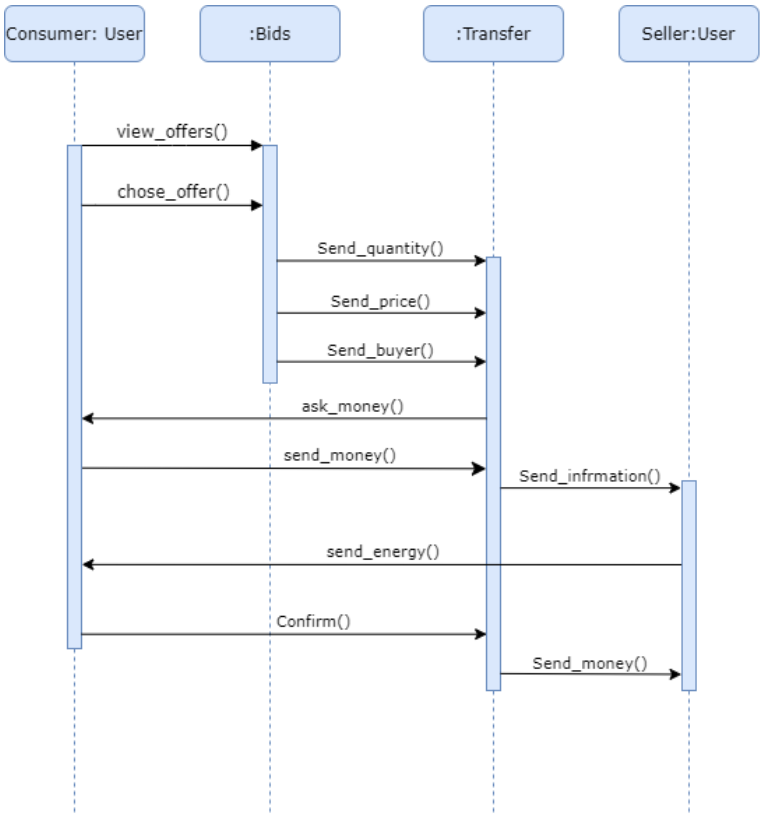


Figure 3.8: Energy transfer example

3.5 Contract Structure

The code was developed in such a way to ensure the highest safety possible for the users. This means that their information was kept apart from other entities that do not have access to them besides the administrator and the user itself. To do this three different contracts were built to better organize the code and divide the distinct functionalities only available to some entities. Figure 3.9 represents relations between different classes in the project in order to clarify the distinction of the code in the chosen pieces.

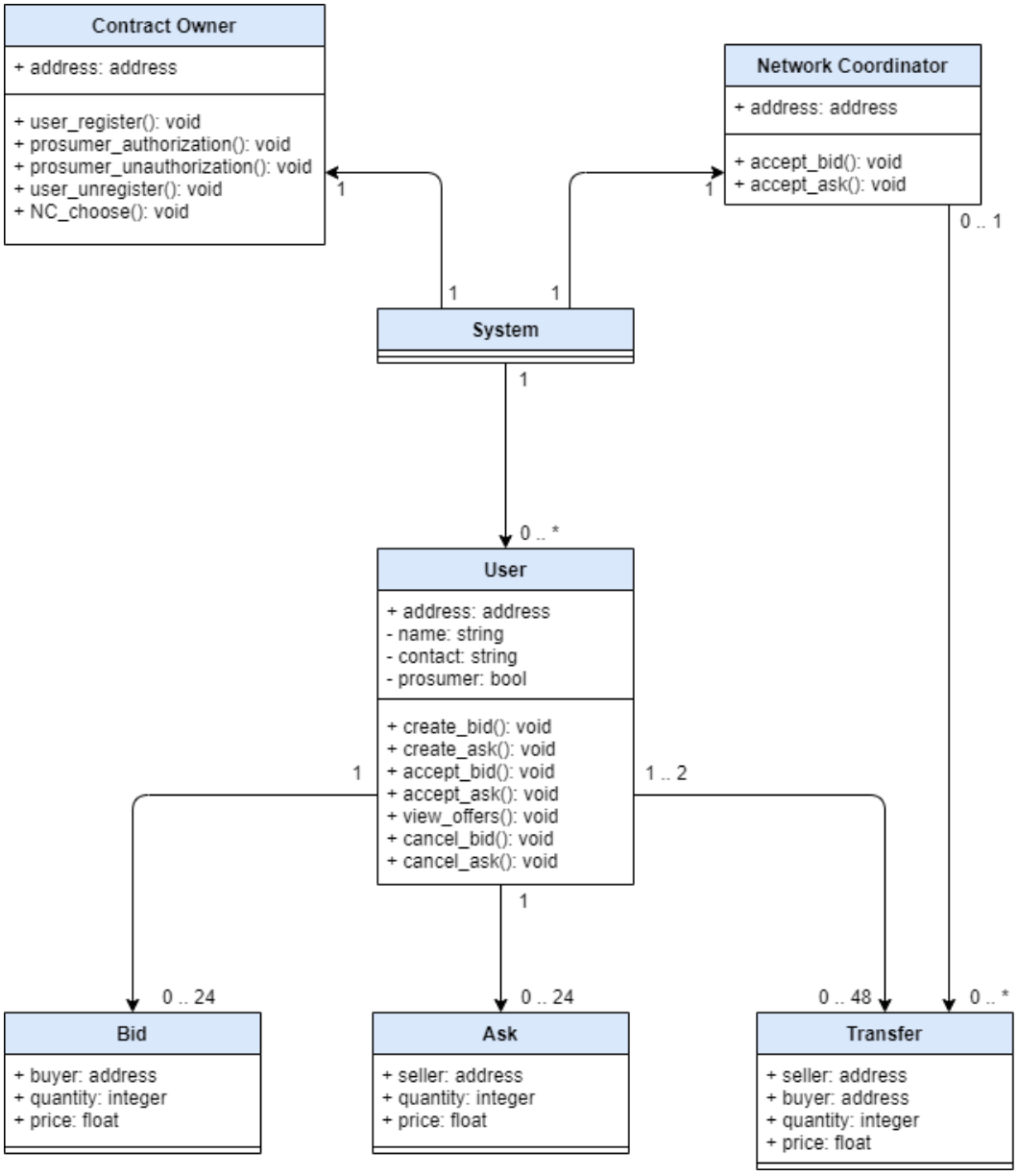


Figure 3.9: System's class diagram

3.5.1 Administrator contract

As soon as the code is executed for the first time and a new instance is created the administrator contract is the first to be executed as it is in this contract that the constructor function resides as shown in Figure 3.10. The only role this function plays is to store the address from the person who ran the contract for the first time in a private variable so the system may remember the user for the next functions that will be needed for some tasks. Beyond this, the contract may also transfer the administrator authority to another address. In the real world, this could be used if the system would be sold. This function is only allowed to be executed by the administrator himself.

```
contract administrator {

    address private service_administrator;
    uint256 private administrator_balance;

    event service_in_execution();
    event new_service_administrator(address service_administrator);

    constructor() public{...}

    modifier only_administrator{...}

    function get_administrator() public view returns (address){...}

    function change_administrator(address new_administrator) public only_administrator{...}
}
```

Figure 3.10: Administrator contract excerpt

3.5.2 User contract

The user contract can be accessed from the administrator's contract, which means that the administrator may enter the functions inside it. In fact, there are functions that only the administrator can execute. It is in this contract that the administrator, and only him, can register and unregister users to the system as well as authorize or remove authorization from a user to be a prosumer. In case access user information is needed it is also inside this contract the administrator may find it since this is where it is stored. All the functions present in this contract are shown in Figure 3.11.

3.5.3 Market contract

The market contract is the most important and the most complex one. Inside this contract, the asks and bids as well the acceptance of those and make of the transfers of energy for money are accessible for both from users and from the administrator. The base price is also calculated inside this contract and may be accessed by every entity. Users are also able to cancel their asks and bids for the period they are no longer needed. Some functions are present in excerpt on the Figure 3.12.

Every function which may require any access to the user's wallet for token transfers shall be signed as payable as its execution requires a monetary exchange. The function which makes the transfer between the users can not be accessed by any entity as is signed as a private function.

This function is only accessed from inside the two functions accept_bid and accept_ask as a way to ensure the data send to the transfer function is not altered after being agreed to by both parties.

```
contract service_user is administrator{

    struct user {...}
    mapping(address => user) public map_users;

    uint number_of_users = 0;
    uint number_of_producers = 0;

    event user_registered(address user);
    event user_deregistered(address user);
    event producer_authorization(address user);
    event producer_deauthorization(address user);

    modifier only_user{...}
    modifier only_producer{...}

    function register_consumer(address person, string memory person_name) public only_administrator {...}

    function deregister_consumer(address person) only_administrator external {...}

    function get_consumer_allowance(address person) external view only_owner returns(bool){...}

    function authorize_producer(address person) only_administrator external{...}

    function deauthorize_producer(address person) only_administrator external{...}

    function get_producer_allowance(address person) external view only_owner returns(bool){...}
}
```

Figure 3.11: Service user contract excerpt

```
contract market is service_user{

    struct order{...}

    order[][24] private list_bid;
    order[][24] private list_ask;

    event bid_created(address buyer,uint256 energy, uint256 price);
    event bid_deleted(address buyer);
    event ask_created(address seller,uint256 energy, uint256 price);
    event ask_deleted(address seller);

    struct transfer{...}

    transfer[] list_transfer;

    function create_bid(uint quantity, uint price, uint period) only_producer external{...}
    function cancel_bid(uint idx, uint period) external{...}
    function delete_bid(uint idx, uint period) private{...}

    function create_ask(uint quantity, uint price, uint period) only_user external{...}
    function cancel_ask(uint idx, uint period) external{...}
    function delete_ask(uint idx, uint period) private{...}

    function accept_bid(uint idx, uint period) only_user external{...}
    function accept_ask(uint idx, uint period) only_producer external{...}
    function make_transfer(address payable buyer, address payable seller, uint energy, uint price) private{...}

    function reset() only_administrator external{...}

    function get_price() external pure returns(uint){...}

}
```

Figure 3.12: Market contract excerpt

3.5.4 Important restrictions

The project has three contracts, as explained. By dividing the functionalities for one contract each, a way to ensure privacy and security is reached. The visibility of variables can also be controlled.

They can be qualified as private or public. These contract variables were made private to ensure they cannot be accessed from outside the contract nor changed. Having one contract for the store of the variable "administrator" which is a private variable makes it impossible for someone outside this contract to access this variable since it can not even see it. The same happens with the user's information which is stored in a different contract to the market operations contract. The market contract can be accessed by the other two.

Some functions are also restricted to some entities also. For example, only prosumers can submit bids on the market, and one way to assure that only those entities execute this function it is possible to view which entity is trying to access it. Solidity eases this verification with the required function. The modifiers are shown in Figure 3.13 which allow the same pieces of code to be used in different functions and help to present a clean code. The same happens for the user register function that can only be executed by the administrator.

```
modifier only_user{
    require(map_users[msg.sender].consumer,'not a user');
    _;
}

modifier only_producer{
    require(map_users[msg.sender].producer,'not a poducer');
    _;
}
```

Figure 3.13: Function's modifiers

3.6 Market Design

In the next section, the steps that lead to the realization of the energy transfers for the next day on the market will be explained. This is a day-ahead market so the orders made on the day are intended to be realized in the next day.

The description starts with the organization of the blocks and orders made for each type of contract. The acceptance of orders leads to a transfer that is the agreement made by two parties but some different information that must be considered.

3.6.1 Time blocks - organization

During the day users may submit their asks and bids according to their allowance. Those orders will be stored in arrays, as previously described, one for each considered period and for each type of order. Users may then see the orders already submitted in the market, all considered for the next day, and accept those which may benefit them the most. Although a user can only submit one type of order for time block he can accept as many orders as he wishes and needs. At the end of

the day, as no more orders may be accepted, all the remaining ones are automatically sent to the Network Coordinator who is now responsible for fulfilling them.

The list of orders is cleared out by the end of the day when the market closes. By this time every order was accepted and there is no need to keep them in the market database. Another way, the amount of information would become some much that the cost to run the program could become unsustainable for the users. Despite this, compound orders are stored in other variables so that previous demands stay registered in order to better forecast future market prices and requirements.

3.6.2 Energy price

It is not economical to store energy so the perfect condition would be to consume the energy as soon as it is produced. When a prosumer is producing energy in excess, he will try to sell that energy for that period of time executing a bid. The prosumer chooses the price he wants to sell his energy for, but he has a base price to better frame his offer with others. The base energy price is calculated taking to account the amount of energy that is being demanded or offered and the prices for each one. This price is not mandatory for the users to execute their offers with, as explained, it only represents a way to assure the prices don't have high fluctuations. The demand for energy is not the same for every time-block so, there is a price for each period.

Taking the example of the Table 3.2 let us assume some bids and asks were made by different users. In this table, amount they want to sell and the price they decided to do it for is represented. The price value chosen is just used as a reference to compare the offers. Bids are then placed in increasing price order and the asks in decreasing price order to develop a market clearing curve. By looking at the graph present on the Figure 3.14 it is easy to visualize that the base price for this period on the market would be 1,10 as that is the price at which the curves intersect.

Table 3.2: Example of offers for a time-block

Offer	User	Quantity (Kw)	Price
Bids	1	12	1
	2	8	1,2
	3	10	1,1
	4	7	0,7
	5	8	0,9
Asks	1	6	0,8
	2	11	1,3
	3	15	1
	4	9	1,1
	5	8	1,2

Prosumers will submit bids for the period under consideration, an amount of energy at a certain price as the consumers submit their asks, the price they are willing to pay for a certain amount of

energy. Bids are ranked according to increasing price and a supply curve is built. The same happens for the demand curve but from the higher to the lower price. The intersection of these two curves is considered the equilibrium price, in this case, the base price for a specific period.

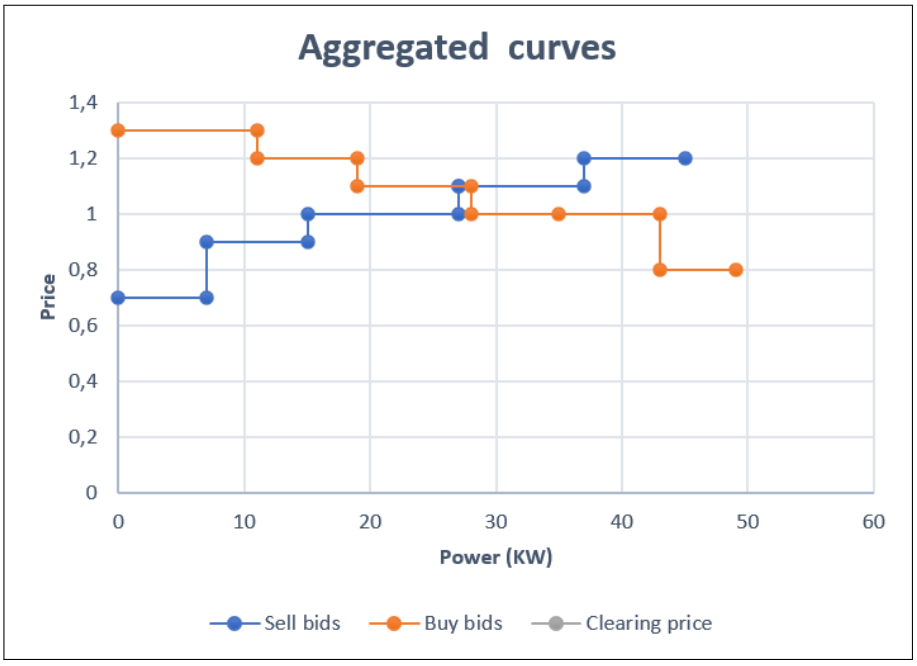


Figure 3.14: Example intersection of curves

The NC operates with his own price with no correlation with the market base price which is fixed the entire time. This price may also vary between time blocks to simulate real time prices for different periods. For periods where the demand is higher, selling price is higher. The Network coordinator's buying price is lower than the executed between peers and the selling price of the NC is higher than the price agreed by the peers.

3.6.3 Transfer of energy for token

When an order is accepted another structure is created. This new structure showing in Figure 3.15 and is used to store the amount of energy the users agreed to exchange and the price set. The address for the buyer and seller is also stored as boolean to sign the transfer as either complete or pending. These structs are then stored in an array. This is the same method used as previously on the orders.

Therefore, looking into a real-world application, as soon as the agreement is made the buyer sends the money to the contract and it stays stored there until he receives the agreed amount of energy. This way neither the seller nor the buyer is able to make a fraudulent action by receiving the money and not transferring the energy or consuming the energy and not paying for it. Money and addresses reside inside a structure which can not be accessed by anyone, not even the administrator. If for any reason the transfer is compromised, for example, the seller runs out of energy,

it is canceled and the money is returned to the buyer. Transfers are also recorded as it could be important to have a list of the previous deal made.

```
struct transfer{
    address buyer;
    address seller;
    uint256 energy;
    uint256 price;
    bool complete;
}

transfer[] list_transfer;
```

Figure 3.15: Code excerpt for transfer storage

3.7 Gas Usage

In this code, some functions consume gas. Some in a low quantity but others may reach a certain gas amount that needs to be considered and is taken into account by the user when executing certain functions.

Some functionalities to be considered are presented below:

- Create a new user or alter prosumer authorization:

  The functions to add a new user to the system and authorize him to be a prosumer are only accessible by the administrator, meaning he is the one who has to pay the fee. Even so, in a real-world project, this would be a fee the user would have to pay the administrator for him to run those functions.

  This was a decision that was taken to find the best way to store this information. The administrator does not have to run a loop to find the user's information. The only thing he needs is the user's blockchain key which is itself a key on the mapping struct.

- Create bids or asks:

  The act of creating a new entry to the orders list is the writing of data to the database. In this way, the entity who executes the function for the order's creation is responsible for paying the fee.

- Energy price calculus:

  A more difficult kind of fee to understand why it must be paid is the one for the calculus of the energy base selling price in the market. This should be accessible for every user to know the mean price being executed at the moment. Otherwise, this value is only visible when

this specific function is executed and since it requires mathematical operations it requires a gas consumption.

- Accept an order:

  The acceptance of an order is itself a new entry to the system. This is the registry of the agreement from those two entities to exchange goods by creating a new entrance to the transfers list. This way there is also a fee which must be paid by the one who accepted the ask or bid.

During this chapter the structure of the code was presented as well as the code developed. It was explained how information was stored inside the system and how to possibly access it. Market organization was mandatory to explain in order to understand how peers would communicate inside the system and exchange needs accords the network. Difficulties were also important to present along the presenting of the functionalities as they were obstacles encountered and overtaken. Following, on the next chapter the system is going to be submitted for testing to prove its efficiency and security for users.

3.8 Chapter summary

During this chapter the structure of the system was presented and discussed. We have introduced the development environment of the system through the system's agents, functionalities that each possess, for example the placement of orders on the market as the structure of the market itself, and how the market would function during a whole day of trading. The store of information was always something to be considered as their access could imply high demand for gas what could become costly for users. This way, some approaches were considered in order to advance in the most efficient way possible.

After presenting the system the next chapter will deal with the tests to validate the functions developed and present some results as well as some issues which were identified during the execution of the code. These errors may disrupt the system's function, efficiency or security and these reasons show why they must be resolved as soon as possible.

Chapter 4

Testing and Results

In this chapter, the results of the tests performed with the code described in the previous chapter of the thesis are presented. This chapter starts by giving an explanation of the environment Remix provides for testing and follows by testing the code function by function that may be encountered in the contracts, from which the most important results will be shown in the next sections.

By the end of this chapter, it will be proven that certain information cannot be accessed by every account present in the network as well as the exchange of goods which are executed as expected. These are thought to be the main characteristics of the code which prove a secure system for its users and an efficient market as well.

4.1 Test Environment

Remix provides its users with the window presented in the Figure 4.1 which will help the developer with running tests for the code. In this window, called Deploy and Run Transactions, which developers can access from the left side menu, some characteristics that may be altered depending on the needs found.

As referred in Chapter 3 on the section about Remix the developer can choose the environment in which he wants to deploy the contracts. The environment chosen, in this case, was JavaScript VM. Every time the developer wants to execute a transaction, he can also choose an account from the list of accounts that are provided, choose a gas limit for the transaction, as well as a value for the gas. In this environment, each transaction is executed immediately, as it is more of a test network and this differs from a real blockchain network. Each account starts with an amount of 100 ether each. Lastly, he may choose the contract on which the transaction will be made. At this time three contracts exist in this project this project: administrator, user and market.

When trying to execute a transaction the user would have to wait until the miner could hash the function and integrate it inside a block. This action would take approximately 15 seconds. This imposed a small limitation on testing but could be easily done if using an injected network provided by a web3 object, for example. For now, the testing is done in this manner as the service won't have many users it is not thought to have too many transactions to execute.

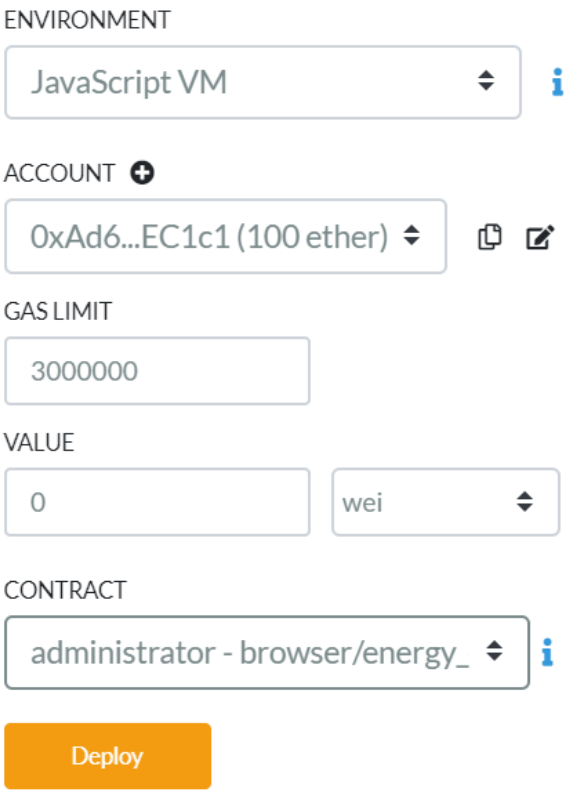


Figure 4.1: Test window

To start the testing, an account to deploy the contract administrator is chosen an account to deploy the contract administrator. As it is the first time the contract will be run, the constructor function will be executed and this account will be stored as the administrator of the system. As with deploying a contract to the network, a transaction a gas limit and gas price may be considered. The standard values used are presented on the Figure 4.1. The next step will be adding five users to the system and assigning half to be prosumers. These users will submit orders to the market at different quantities and prices, simulating a real example. Users will then look for offers that best fulfill their needs. At a certain time, no more offers are accepted and the market closes and remaining orders are automatically accepted by the NC. At the end of the testing pahse, costs will be calculated for consumers that made agreements with prosumers or with the NC also.

4.2 Administrator

Deploying the administrator contract for the first time, as described, with the account presented in the Figure 4.1 the system is executed with a new administrator. In the terminal a message is received, which is present in Figure 4.2, informing the developer with some information about the successful transaction that was just just executed. In this message we can see some important

details such as the successful status, the hash codes of both the transaction and of the block in which the transaction was included, the address of the contract and also the sender and receiver addresses to whom the transaction was destined. This message also presents the developer with the consumed gas. Transaction cost was of 238.664 gas units from which 138.608 were execution costs of the transaction. The consumed gas was inferior to the set limit and so it executed.



Figure 4.2: Successful transaction

Inside the window present in Figure 4.1 it is received the following sub-window, present in Figure 4.3. On this window, the developer receives access to execute the functions inside this contact. In this case, there is a function present to pass the administration of the system to another account, which only can be executed by the system's administrator itself and another to get the address of the current administrator.

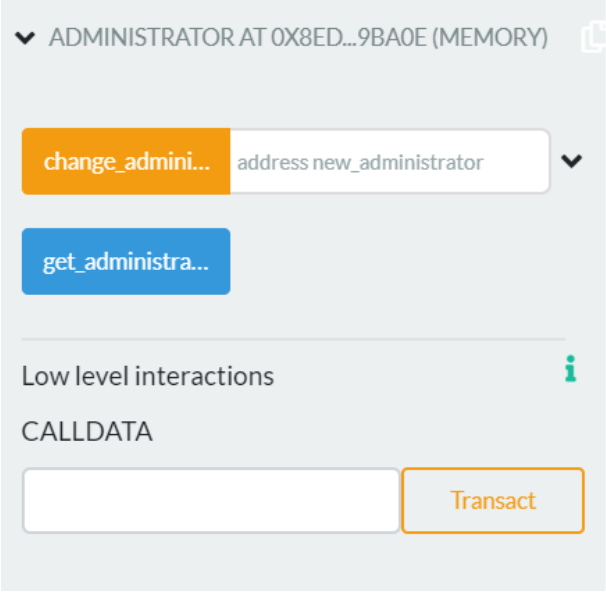


Figure 4.3: Administrator contract

Before continuing with the tests the other two contracts must be deployed. The same account is used for that purpose. When deploying each one of these contracts we receive a message informing about the result of the transaction. We highlight the gas consumed of 1.419.253 to deploy the service_user contract, which is still below to the 3.000.000 limit.

When trying to deploy the market contract it was received an error showing in Figure 4.4. The gas limit was reached, the reason why the transaction could not be finalized but the limit of 3.000.000 still has to be paid by the administrator. For this reason, the gas limit was altered to 4.000.000 and another attempt was realized, this time successfully with a consumption of 3.018.453 gas units.

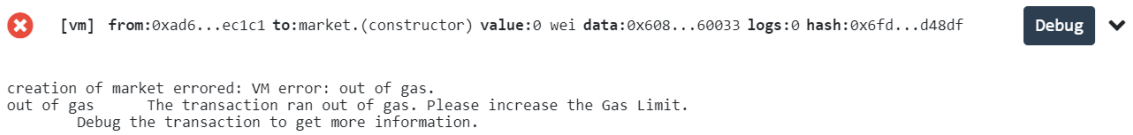


Figure 4.4: Gas limit reached

4.3 User contract

After deploying the service_user contract a window like the one for the administrator contract is received. This window is showed in Figure 4.5. In this, all the functions present inside this contract are accessible in addition to the functions present in the previous contract since it is passed to this contract so that it can recognize the system's administrator.

The new functions in this contract are to register and unregister a user, allow a user to be a prosumer or take this allowance and finally, three functions to access users information, for example, their allowance to produce or consume energy from the market.

4.3.1 Register users

To register new users in the system we will use the addresses linked to the administrator address in the accounts provided. This function can only be executed by the system's administrator and so the transaction must be made by him. To execute this function, we need the user's address and name. When registering him to the network we will automatically store 100 units of energy and he is considered solely a consumer not a prosumer. Registering a new user to the system we receive a new message similar to the previous messages every time we executed a transaction. The transaction consumed a total of 117.952 gas units.

Now we have our first user in the system and more users be added to ensure there are enough users to trade energy within the system. Before that, we show how only the administrator can add users to the system. We do this by trying to add a new user executing the transaction with the address of the first user, instead of address of the system administrator. The terminal shows the

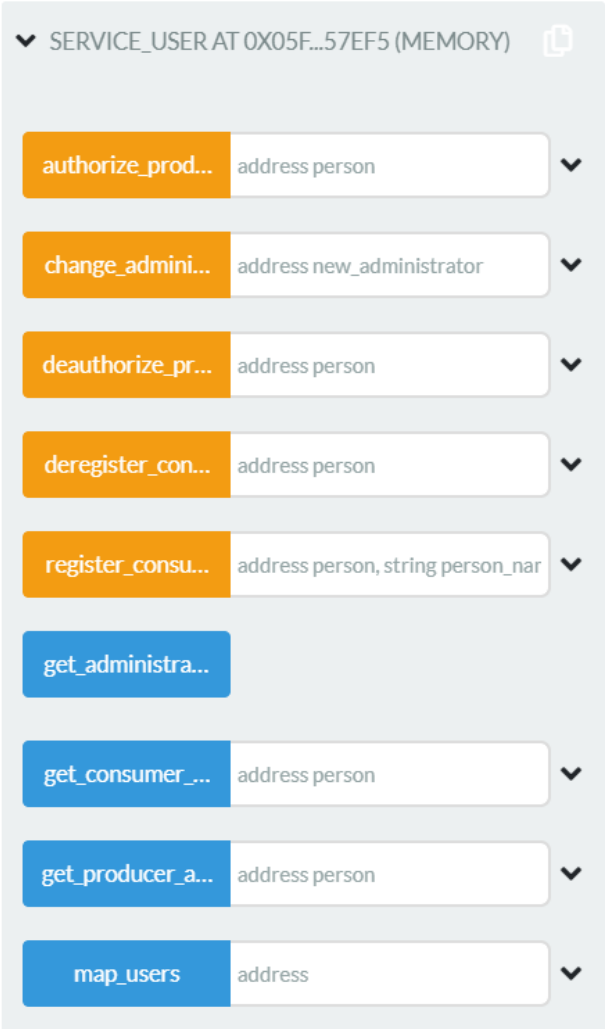


Figure 4.5: User contract deployed

error message present in Figure 4.6. The transaction was not finished but it still has a cost for the account who tried to execute it. The transaction consumed a total of 24.990 gas units and could not add any information to the system. To have a more complete system more 4 users will be added, this time correctly.

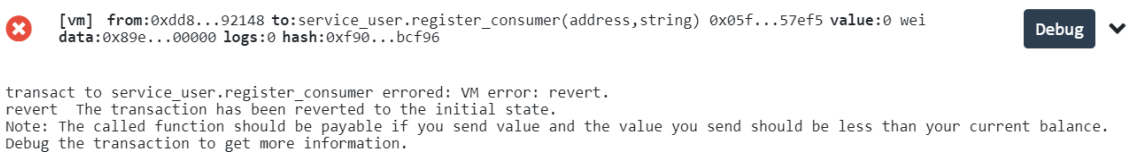


Figure 4.6: Trying to add a new user by other address than the administrator

4.3.2 Authorize a prosumer

The system needs both consumers and prosumers to be completed, so the next step will be to modify three accounts to allow them to be prosumers. To run this function the only parameter needed is the user address. The system will access the specific structure from the mapping and will change the boolean prosumer to be true. The address passed must be present on the list of users as the address that runs the function must be the administrator. If the administrator tries to run this function with another address that is not registered he will receive a message like this present in Figure 4.7. The message clearly states that the address is not a registered user.

Following with the registry of users, changing the first three users to be prosumers has a consumption of 53.752 gas units.

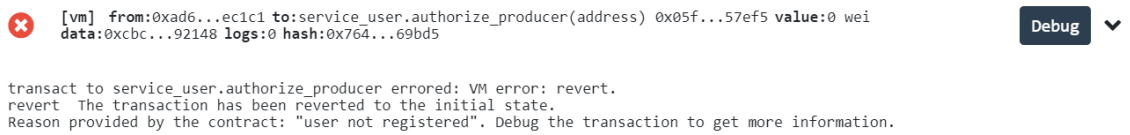


Figure 4.7: Trying to give allowance to a not registered address

The structure of a prosumer through the mapping the data is shown in Figure 4.8. Here we can see the variables stored as name and two Boolean strings indicating if the user ia a consumer and/or prosumer which in this example are both set to true. If the administrator wants to revoke this allowance he must change these variables to false, taking to account that a prosumer must always be a registered user.

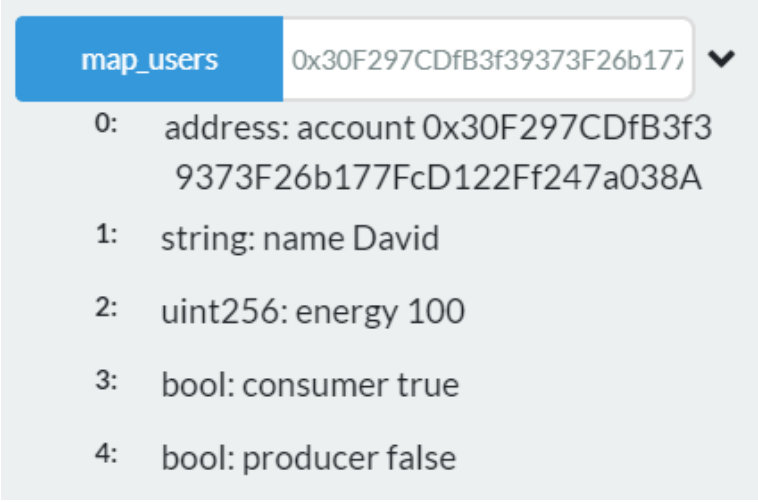


Figure 4.8: Looking into a prosumer structure

4.3.3 Network Coordinator

The Network Coordinator is for now stored only as an address and an amount of energy inside the same contract were the users are stored. This address is also provided by Remix and as the same amount of ether as any other user to the system. This entity may execute more transactions compared to other users but the amount of tokens each has is enough for this case study. In this example, there is no need to show any message received or code executed as there is no difference to the previously executed functions. This inclusion of a new entity is only available to the administrator as previously to the users and there is a possibility to choose another account to have these functions if intended.

All registered entities are shown in Table 4.1 from which it is possible to see their Ethereum account addresses as well as system's allowances and names in the case of users.

Table 4.1: System's tests entities

Address	Name	Entity
0xAd6...EC1c1	-	Administrator
0x236...8c5E4	John	Prosumer
0x8a9...222c3	Maria	Prosumer
0x30F...a038A	David	Prosumer
0x940...31dA4	Joana	Consumer
0x236...8c5e4	Simon	Consumer
0xf47...d6c20	Emily	Consumer
0x359...4c98b	-	NC

4.4 Market contract

After deploying the market contract, a similar Remix window to the other contracts is available to access its functions. This time all the functions are present in this window as the other two contracts are passed to this one so that the information stored in the previous ones is also available in this market contract. The quantity of functions in this window makes it very extensive difficult to capture all of the information in one figure.

New functions in this contract are the ability to execute and cancel orders as well as the ability to accept them and make a transfer between two registered entities, see offers, reset the market for the next day and get the base price. The two types of orders, asks and bids, share the same structure as both have a related account an amount of energy and a price to store. Even though, they are executed in different functions as these structures are stored in different arrays and have to check some parameter to be executed, for example, who is running the transaction.

4.4.1 Asks

To submit an ask the account must be a registered user to the system. The NC or the administrator cannot execute this type of contract. When trying to submit a new order with the administrator

address the terminal shows the message presented in Figure 4.9. We can see who made the transaction by looking at the data in the Figure where the administrator's address is showing. Prosumers may also submit asks as they are registered users to the system.

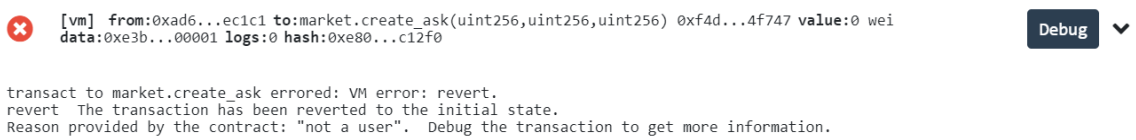


Figure 4.9: Trying to execute an order with admin account

When executing an ask with a consumer account, a message of a successful transaction is received from which an excerpt is present in Figure 4.10. On this image it is possible to see the account who executed the transaction, the name of the event, and the variables chosen by them. The creation of a new ask as a total of 108.486 consumed gas units.



Figure 4.10: Creating an ask

If for any reason a user wants to cancel his ask, the system gives them that ability but a fee needs to be paid of 29.631 gas units. If for any reason a user tries to delete another user's ask he will receive the following message present in Figure 4.11.

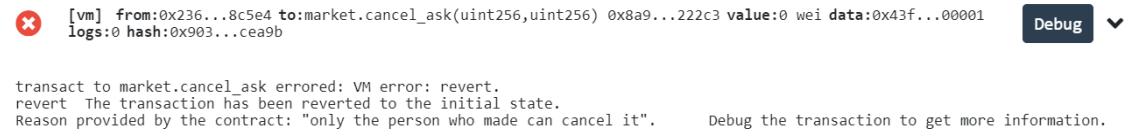


Figure 4.11: Trying to delete an ask of another account

4.4.2 Bids

Submitting a bid works in the same way as the ask. The only entity that is allowed to make them are producers who need a quantity, a price and a period to which they want to submit the offer. If

any consumer wants to execute this transaction he will receive the message present in Figure 4.12 saying that he is not registered as a prosumer.

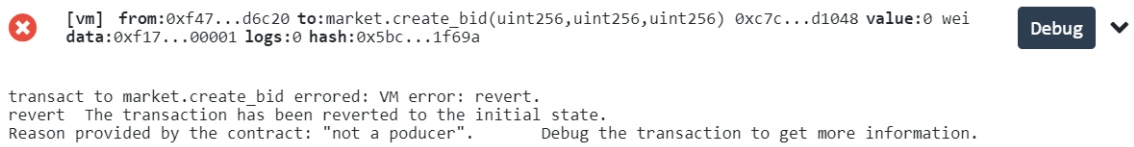


Figure 4.12: Trying to execute an order with a consumer account

When an authorized prosumer wants to submit a bid to the market, he will receive a similar message to the one showing on the ask subsection. This message present in Figure 4.13 wich shows the event realized and the information written inside the new structure created stored inside the specific string related to the kind of order and period. The total consumed gas for this transaction is 108.612 units.



Figure 4.13: Creating a bid

As for the example of an ask, a prosumer may also delete his bid if he wants but again there is a fee of 29.642 units. If a user tries to delete another prosumer's bid he will receive a message like the previous showed, present in Figure 4.14.

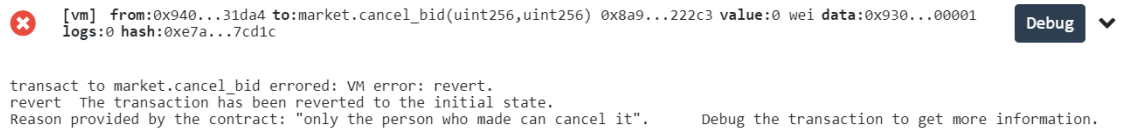


Figure 4.14: Trying to delete a bid of another account

4.4.3 Accept an order

After looking for offers on the market the users may choose any that may please them. One thing that must be considered is that only registered users may accept any orders.

When accepting an order there are two different functions to call: one for accepting bids and another to accept asks. These two functions will call the same one putting the parameter on the intended order. This function stores the transfer and represents the agreement between the two

peers. Its parameters are the buyer's address, seller's address, the quantity of energy, and price for that quantity. This function is private and can only be accessed from other contract functions as it is not visible to users. Its functionality resides on storing the values in a specific structure, adding a boolean as a flag to check if it was completed or not. When an order is accepted the boolean that checks if the order was accepted is turned to true so that it cannot be accepted by any other user as it does not appear in the show function.

For test purposes, the transfer of energy is automatically executed as the transfer of tokens and the transfer is automatically considered as closed.

As explained, there are some limitations when accepting orders. The easiest to detect is that only registered users may accept any kind of offers. When a non-registered account tries to accept a bid, for example, it receives a message like the one present in Figure 4.15 informing them that they are not users. On the other hand, consumers only can accept bids and are not allowed to accept asks. For this reason, when trying to execute such transaction they will receive the message present in Figure 4.16 informing he is not a registered producer.

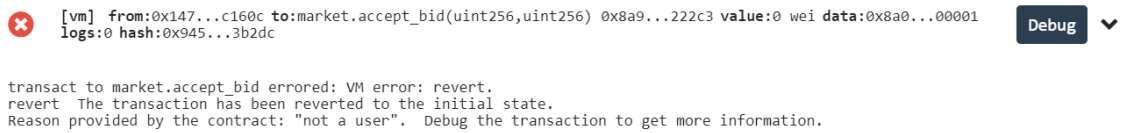


Figure 4.15: Trying to accept a bid with a non-user account

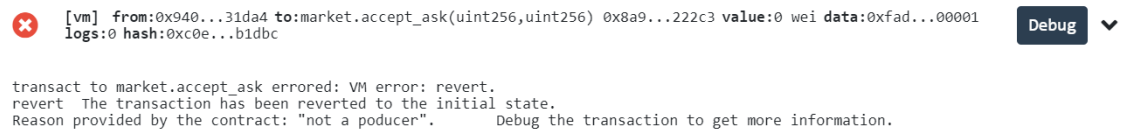


Figure 4.16: Trying to accept an ask with a consumer

Trying to accept a bid with a registered user, he receives the message shown in Figure 4.17. From this message is possible to know the status of the transaction, who is the emitter, and the receiver of the transaction. The receiver of the message is not the buyer, as it could be thought, the transaction is stored in the contract and so, it is the message receiver. Accepting a bid consumes a total of 168.150 gas units as can be seen in the image. Yet, accepting an ask consumes a bit less, making a total of 136.740 gas units.

For example, consider two users of the system, a common user, and a prosumer. The prosumer set the bid presented in Figure 4.13 which was accepted from the user and received the message shown in Figure 4.17. As it is possible to see the total amount of energy transferred was of 10 units and the price 1 unit. In the next two figures it is possible to see the structures of the two users, the prosumer in Figure 4.18 and the consumer in Figure 4.19. Considering that both started that simulation with an amount of 100 energy units it is clear to see that there was a transfer from the first to the second of 10 units.

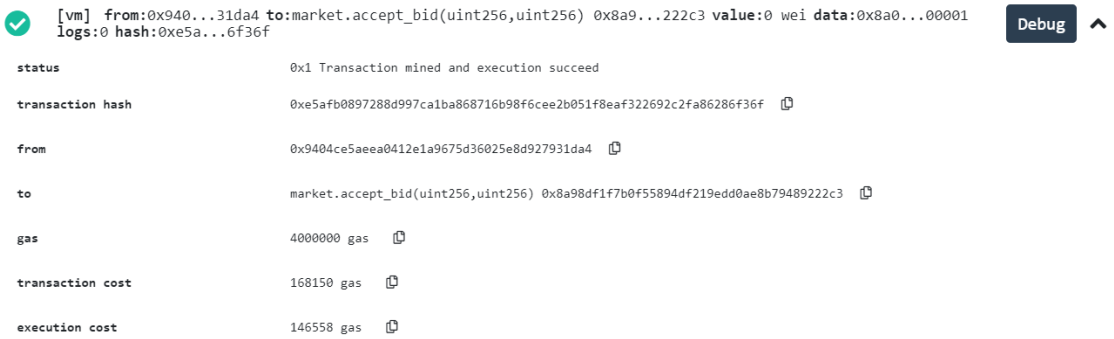


Figure 4.17: Successful acceptance of a bid

0: address: account 0x2367Cf6E50810d5c5Ece8a69232AC2ca6938c5E4

1: string: name John

2: uint256: energy 90

3: bool: consumer true

4: bool: producer true

Figure 4.18: Seller information

0: address: account 0x9404CE5AeeA0412e1A9675D36025e8D927931dA4

1: string: name Joana

2: uint256: energy 110

3: bool: consumer true

4: bool: producer false

Figure 4.19: Buyer information

4.4.4 Clearing of market information

Before the end of the day, the market has to be cleared and the offers that were not accepted by the users must be delivered to the NC so that there is no user left aside. This action is made by the administrator by running a function that automatically executes the transfers between users and the NC as well as deletes all offers from the market. This way all the lists are empty for the next day.

4.5 Cost Benefits

Simulating the case study first described in the beginning of this chapter, where there are a total of 5 asks and 5 bids for the same period placed by the registered users. These are the offers previously presented in Chapter 3 but follow and are summarized in Table 4.2. As calculated before the base price for this period with these orders is 1,1 units of price.

Table 4.2: Executed orders on the market

Asks (consumers buying)		Bids (prosumers selling)	
Quantity ask	Price ask	Quantity bid	Price bid
11	1,3	7	0,7
8	1,2	8	0,9
9	1,1	12	1
15	1	10	1,1
6	0,8	8	1,2

Looking at the information presented in this table it is possible to confirm that not all the offers will be accepted by the users since they may have a higher price or ask/offer an amount of energy which is not needed in other offers. Table 4.3 shows the amounts of energy traded between users and with the NC. Consumers will have to trade a total of 21 units with the NC and prosumers a total of 24 units of energy. The total amount of energy traded between peers is of 28 units.

Table 4.3: Traded offers among users and NC

-	Quantity	Price
Market Clearing	28	1,1
Unserved asks	21	NC selling price (above 1,1)
Unserved bids	24	NC buying price (below 1,1)

Considering the prices of the energy simulated in this exercise, there must be a assumption that the NC acts as a price taker. He is obligated to accept offers from users but they are obligated to accept the prices that he practices. A comparison between a transitional market and the developed Peer-to-Peer market is shown in Table 4.4 which shows the prices of consumers if they could only buy energy from the NC and the prices they paid in this market. For this use case, it is assumed that NC buys energy for 0,5 units of price and sells for 1,5.

Table 4.4: Costs and savings for users

Normal Market		P2P Market	
Entity	Transfers	Entity	Transfers
Consumers	- 73,5	Consumers	- 62,3
Prosumers	+ 22,5	Prosumers	+ 42,8

As it is possible to see there is a total of money saved in this study. Consumers had a reduction of costs in 11,2 units of price as well as consumers who increased their gains in 20,3 unit of price. This equates to a reduction in cots of 15.2% in energy costs for consumers and an increase in 90.2% in revenues for prosumers compared to the traditional market structure.

Lastly, we have only to consider the prices of fees paid during the operation of the smart contracts. Throughout this chapter, gas values consumed during their execution were presented. The prices in ETH of these consumptions were not presented because each user when trying to execute a transaction, has to specify a value he is willing to pay for the amount of gas that function is going to consume. In Table 4.5 there is a summary the amount of consumed gas of each important function of the system, as well as a price paid if the price per unit was 38 Gwei and the entity who would pay for this fee. These costs come in ether but is possible to confirm that they have a low value when compared to administration prices we may encounter in real-world applications. Using a exchange rate of 202 Euros per ETH, it is possible to compare these costs so that the reader can have a term of comparison.

Table 4.5: Fees costs

Function	Consumed gas	price (ETH)	price (EUR)	payer
Deploy administrator contract	138.608	0.00000527	0,0011	Administrator
Deploy user contract	1.419.253	0.00005393	0,011	Administrator
Deploy market contract	3.018.453	0.00011470	0,023	Administrator
Register new user	117.952	0.00000448	0,00090	Administrator
Authorize prosumer	53.752	0.00000204	0,00041	Administrator
Submit ask	108.486	0.00000412	0,00083	Consumer
Submit bid	108.612	0.00000413	0,00083	Prosumer
Accept ask	136.740	0.00000520	0,0011	Seller
Accept bid	168.150	0.00000639	0,0013	Buyer

4.6 Chapter summary

This chapter presented the validation of the most important functionalities of this project. Starting with the first deployment of the code and always taking to account the addresses that were trying to execute the transactions. Gas usage was also an important measurement to be pointed out as it was shown, since limits on its payment may stop miners from proceeding with the execution of the function. We showed that security was an essential objective to be reached. For this reason it was shown what would happen unauthorised accounts tried to access users information as well as their offers. Concluding testing we are able to reach some outcomes for this project, verify if the main objectives were reached, draw conclusions and identify some possible next steps and these will be discussed in the following chapter.

Chapter 5

Conclusions and Future Works

This chapter summarises the contributions and draws conclusions of the work undertaken for this thesis. The potential impact of the blockchain technology on residential consumers has been discussed at length and this chapter highlights the main areas where blockchain can play a significant role in the future energy system. Building upon these conclusions, a few ideas for the future extension of the model are presented and discussed.

5.1 Conclusions

It started by explaining the present conditions the world is in regarding the market structure and this new technology, the blockchain, which is entering new applications, including the energy sector. The energy market was detailed as an outdated system, which is still constructed similarly as it was 100 years ago. Yet, it currently faces huge development that came with the introduction of renewable energy sources, especially in home appliances, and IoT devices which allowed a digitalization of the system and consequently brought easiness of communications between producers, distributors, and consumers. The last big revolution may come with new technologies like blockchain that may construct a stabler and more secure system, taking power from big producers and bringing it to consumers, making a decentralized system.

Blockchain was introduced as a disruptive technology, able to change the way we do current day-to-day activities such as banking or gambling. The first application of blockchain, Bitcoin, was referred to as a money transfer opportunity for lower fees in transfers and the ability to dismiss banking needs for these purposes. Ethereum saw something that Bitcoin failed to fulfill and brought the ability to construct virtual agreements from two network users, the smart contracts. This rapidly was seen as an opportunity for different industries that started constructing their blockchains. Smart contracts were the main focus on this project as they worked as a way for users to make a promise to send money for excess production across the network.

After introducing the energy market and blockchain technology we started to develop the system that allowed the Peer-to-peer market. The code was presented always regarding user security of information, energy, and money. The main functions were described in a focused way so that

the reader could identify difficulties encountered during the development of this work. Since this is a relatively new technology which is still suffering huge improvements it may be difficult to find help even on digital resources.

In conclusion, through the development of this Remix model, the main objectives of this thesis have been achieved. This was done by creating a blockchain based energy trading platform which placed consumers at the heart of the system and allowed them to trade energy directly with each other. As this system relied upon blockchain, it was safe and secure and protected the data of the users. There were measures put in place to ensure that only the originators of a transaction could edit the transaction data which ensured that the correct quality of energy as well as the correct amount of money was transferred. The role of the external network coordinator was reduced which led to cost savings for the consumer while still having the coordinator as a supplier of last resort to ensure the safety and stability of the system.

From the simulations performed, it was also possible to see an improvement in the user's financial situation. Costs were reduced for consumers and an increase in profit was visible in the side of prosumers. Although not all the offers could match in the test case this is not a major issue as there is not always complete matching in real world markets. An increase in the number of accepted offers would provide better results for both the consumers and prosumers and this could be achieved through a larger network of peers.

5.2 Future Work

Despite the advances made during this project, there are some areas that could be improved before this system is tested in the real world. The first improvement would be to simulate the program with a higher number of users and in a restrictive network that limits the number of transactions per second the system can execute. This would make a realistic approach of real network projects need to be deployed in so they can be accessed by the general public.

Another feature that will be addresses in future work will be the autonomous market. After users send their orders to the market their bids and asks would be automatically linked in matching pairs which would respect quantities and prices specifies by senders. This would reduce user interactions in the system and could even help them to find matching offers and possibly even reduce costs or increase profit for prosumers by reducing the need for the Network Coordinator.

References

[1] Bharat Bhushan, Aditya Khamparia, K. Martin Sagayam, Sudhir Kumar Sharma, Mohd Abdul Ahad, and Narayan C. Debnath. Blockchain for smart cities: A review of architectures, integration trends and future research directions. *Sustainable Cities and Society*, 61:102360, Oct. 2020. URL: https://doi.org/10.1016/j.scs.2020.102360.

[2] Matthew Gough, Rui Castro, Sérgio F. Santos, Miadreza Shafie-khah, and João P.S. Catalão. *Chapter 2 - A panorama of applications of Blockchain technology to energy*. Miadreza Shafie-khah, Blockchain-based Smart Grids, Academic Press, 2020. URL: https://doi.org/10.1016/B978-0-12-817862-1.00002-6.

[3] David Livingston, Madison Freeman, Varun Sivaram, and Maximilian Fiege. Applying blockchain technology to electric power systems. *Smart Energy International*, pages 95–100, Oct.

[4] Francisco López-Ramos, Stefano Nasini, and Mohamed H. Sayed. An integrated planning model in centralized power systems. *European Journal of Operational Research*, 287:361–377, November 2020. URL: https://doi.org/10.1016/j.ejor.2020.05.006.

[5] Juhar Abdella and Khaled Shuaib. Peer to peer distributed energy trading in smart grids: A survey. *Energies*, 11:1560, June 2018. URL: https://doi.org/10.3390/en11061560.

[6] Rana.H.A. Zubo, Geev Mokryani, Haile-Selassie Rajamani, Jamshid Aghaei, Taher Niknam, and Prashant Pillai. Operation and planning of distribution networks with integration of renewable distributed generators considering uncertainties: a review. *Renew Sustain Energy Rev*, 72:1177–1198, May 2017. URL: https://doi.org/10.1016/j.rser.2016.10.036.

[7] Yu Han, Yingying Sun, and Junjie Wu. An efficient solar/lignite hybrid power generation system based on solar-driven waste heat recovery and energy cascade utilization in lignite pre-drying. *Energy Conversion and Management*, 205:112406, Feb. 2020. URL: https://doi.org/10.1016/j.enconman.2019.112406.

[8] R. Elankovan, Krishnadass Karthick S. Suresh, Mohammed Muaaz M.D. Hussain, and V.P. Chandramohan. Evaluation of thermoelectric power generated through waste heat recovery from long ducts and different thermal system configurations. *Energy*, 185:477–491, Oct. 2019. URL: https://doi.org/10.1016/j.energy.2019.07.039.

[9] Quan Yang, Riccardo Dettori, Ganjun Yuan, and Luke R. Anderson. A perovskite solar cell owing very high stabilities and power conversion efficiencies. *Solar Energy*, 201:541–546, May 2020. URL: https://doi.org/10.1016/j.solener.2020.02.085.

[10] Sha Xue, Chuting Tan, Praneeth Kandlakunta, Ibrahim Oksuz, Vasil Hlinka, and Lei R. Cao. Methods for improving the power conversion efficiency of nuclear-voltaic batteries. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 927:133–139, May 2019. URL: https://doi.org/10.1016/j.nima.2019.01.097.

[11] A. Datas, A. Ramos, and C. del Cañizo. Techno-economic analysis of solar pv power-to-heat-to-power storage and trigeneration in the residential sector. *Applied Energy*, 256:113935, Dec. 2019. URL: https://doi.org/10.1016/j.apenergy.2019.113935.

[12] Simone Tagliapietra, Georg Zachmann, Ottmar Edenhofer, Jean-Michel Glachant, Pedro Linares, and Andreas Loeschel. "the european union energy transition: Key priorities for the next five years". *Energy Policy*, 132:950–954, Sep. 2019. URL: https://doi.org/10.1016/j.enpol.2019.06.060.

[13] European Commission. 2030 climate energy framework. URL: https://ec.europa.eu/clima/policies/strategies/2030_en [last accessed 18.06.2020].

[14] European Commission. 2050 long-term strategy. URL: https://ec.europa.eu/clima/policies/strategies/2050_en#tab-0-0 [last accessed 18.06.2020].

[15] Roberto Bocca, Pedro G. Gómez Pensado, Bandar Alkhamies, Thomas Seitz, Nikhil Patel, Kassia Yanosek, and Ann Hewitt. "game changers in the energy system emerging themes reshaping the energy landscape". *World Economic Forum*, pages 8–9, Jan. 2017. URL: http://www3.weforum.org/docs/WEF_Game_Changers_in_the_Energy_System.pdf.

[16] F. Simon. The market for demand-response is only going to grow. URL: https://www.euractiv.com/section/energy/interview/energy-execsthe-market-for-demand-response-is-only-going-to-grow/ [last accessed 27.02.2020].

[17] Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. Designing microgrid energy markets. *Applied Energy*, 210:870–880, Jan. 2018. URL: https://doi.org/10.1016/j.apenergy.2017.06.054.

[18] Maria Luisa Di Silvestre, Salvatore Favuzza, Eleonora Riva Sanseverino, and Gaetano Zizzo. How decarbonization, digitalization and decentralization are changing key power infrastructures. *Renewable and Sustainable Energy Reviews*, 93:483–498, Oct. 2018. URL: https://doi.org/10.1016/j.rser.2018.05.068.

[19] Ricardo Moura and Miguel Centeno Brito. Prosumer aggregation policies, country experience and business models. *Energy Policy*, 132:820–830, Sep. 2019. URL: https://doi.org/10.1016/j.enpol.2019.06.053.

[20] Ciprian Nemes and Florin Munteanu. Computer typesetting of technical journals on UNIX. In *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, pages 37–41, Lasi, Romania, Oct. 2018. URL: https://doi.org/10.1109/ICEPE.2018.8559854.

[21] EDP. URL: https://www.edp.pt/particulares/apoio-cliente/perguntas-frequentes/pt/leituras/contadores-inteligentes/

o-que-e-um-contador-inteligente-e-como-funciona/faq-11235 [last accessed 2020-04-03].

[22] Shaoyuan Yu, Shengchun Yang, Yaping Li, and Jian Geng. Distributed energy transaction mechanism design based on smart contract. In *2018 China International Conference on Electricity Distribution (CICED)*, pages 2790–2793, Tianjin, China, Sep. 2018. URL: https://doi.org/10.1109/CICED.2018.8592130.

[23] Richard Hanna, Matthew Leach, and Jacopo Torriti. Microgeneration: The installer perspective. *Renewable Energy*, 116:458–469, Feb. 2018. URL: https://doi.org/10.1016/j.renene.2017.09.023.

[24] Renata I.S. Pereira, Sandro C.S. Jucá, and Paulo C.M. Carvalho. Iot embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants. *Measurement*, 142:195–212, Aug. 2019. URL: https://doi.org/10.1016/j.measurement.2019.04.085.

[25] Rehman Zafar, Anzar Mahmood, Sohail Razzaq, Wamiq Ali, Usman Naeem, and Khurram Shehzad. Prosumer based energy management and sharing in smart grid. *Renewable and Sustainable Energy Reviews*, 82:1675–1684, Feb. 2018. URL: https://doi.org/10.1016/j.rser.2017.07.018.

[26] S. Sofana Reka and Tomislav Dragicevic. Future effectual role of energy delivery: A comprehensive review of internet of things and smart grid. *Renewable and Sustainable Energy Reviews*, 91:90–108, Aug. 2018. URL: https://doi.org/10.1016/j.rser.2018.03.089.

[27] Mark A. Jamison and Palveshey Tariq. Five things regulators should know about blockchain (and three myths to forget). *The Electricity Journal*, 31:20–23, Nov. 2018. URL: https://doi.org/10.1016/j.tej.2018.10.003.

[28] Merlinda Andoni, Valentin Robu, David Flynn, Simone Abram, Dale Geach, David Jenkins, Peter McCallum, and Andrew Peacock. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100:143–174, Feb. 2019. URL: https://doi.org/10.1016/j.rser.2018.10.014.

[29] A. Paaso M. Yan e M. Shahidehpour Z. Li, S. Bahramirad. *Blockchain for decentralized transactive energy management system in networked microgrids, Electr. J., vol. 32, n. 4, pp. 58–72*. Mai. 2019.

[30] SolarCoin. Solarcoin. URL: https://solarcoin.org/ [last accessed 01.06.2020].

[31] The Sun Exchange. About sun exchange. URL: https://thesunexchange.com/about-us [last accessed 01.06.2020].

[32] A. Ahl, M. Yarime, M. Goto, Shauhrat S. Chopra, Nallapaneni Manoj. Kumar, K. Tanaka, and D. Sagawa. Exploring blockchain for the energy transition: Opportunities and challenges based on a case study in japan. *Renewable and Sustainable Energy Reviews*, 117:109488, Jan. 2020. URL: https://doi.org/10.1016/j.rser.2019.109488.

[33] Arne Meeuw, Sandro Schopfer, Anselma Wörner, Verena Tiefenbeck, Liliane Ableitner, Elgar Fleisch, and Felix Wortmann. Implementing a blockchain-based local energy market: Insights on communication and scalability. *Computer Communications*, 160:158–171, July 2020. URL: https://doi.org/10.1016/j.comcom.2020.04.038.

[34] Antonio J. Conejo and Ramteen Sioshansi. Rethinking restructured electricity market design: Lessons learned and future needs. *International Journal of Electrical Power Energy Systems*, 98:520–530, June 2018. URL: https://doi.org/10.1016/j.ijepes.2017.12.014.

[35] Andrea Dertinger and Lion Hirth. Reforming the electric power industry in developing economies evidence on efficiency and electricity access outcomes. *Energy Policy*, 139:111348, April 2020. URL: https://doi.org/10.1016/j.enpol.2020.111348.

[36] J. Lowitzsch, C.E. Hoicka, and F.J. van Tulder. Renewable energy communities under the 2019 european clean energy package – governance model for the energy clusters of the future? *Renewable and Sustainable Energy Reviews*, 122:109489, April 2020. URL: https://doi.org/10.1016/j.rser.2019.109489.

[37] João Crispim, José Braz, Rui Castro, and Jorge Esteves. Smart grids in the eu with smart regulation: Experiences from the uk, italy and portugal. *Utilities Policy*, 31:85–93, Dec. 2014. URL: https://doi.org/10.1016/j.jup.2014.09.006.

[38] European Union. Clean energy for all the europeans. pages 12–13, 2019.

[39] Sana Noor, Wentao Yang, Miao Guo, Koen H. van Dam, and Xiaonan Wang. Energy demand side management within micro-grid networks enhanced by blockchain. *Applied Energy*, 228:1385–1398, Oct 2018. URL: https://doi.org/10.1016/j.apenergy.2018.07.012.

[40] Michel Noussan and Simone Tagliapietra. The effect of digitalization in the energy consumption of passenger transport: An analysis of future scenarios for europe. *Journal of Cleaner Production*, 258:120926, June 2020. URL: https://doi.org/10.1016/j.jclepro.2020.120926.

[41] Tiago Sousa, Tiago Soares, Pierre Pinson, Fabio Moret, Thomas Baroche, and Etienne Sorin. Peer-to-peer and community-based markets: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 104:367–378, April 2019. URL: https://doi.org/10.1016/j.jclepro.2020.120926.

[42] Christine Eon Gregory M. Morrison Björn Sandén Sam Wilkinson, Kristina Hojckova. Is peer-to-peer electricity trading empowering users? evidence on motivations and roles in a prosumer business model trial in australia. *Energy Research Social Science*, 66:101500, Aug. 2020. URL: https://doi.org/10.1016/j.erss.2020.101500.

[43] Thurner Thomas Wolfgang Brilliantova Vlada. Blockchain and the future of energy. *Technology in Society*, 57:38–45, May 2019. URL: https://doi.org/10.1016/j.techsoc.2018.11.001.

[44] Dongdong Yue, Ruixuan Li, Yan Zhang, Wenlong Tian, and Yongfeng Huang. Blockchain-based verification framework for data integrity in edge-cloud storage. *Journal of Parallel and Distributed Computing*, page S0743731520303142, June 2020. URL: https://doi.org/10.1016/j.jpdc.2020.06.007.

[45] Bin Cao, Zhenghui Zhang, Daquan Feng, Shengli Zhang, Lei Zhang, Mugen Peng, and Yun Li. Performance analysis and comparison of pow, pos and dag based blockchains. *Digital*

Communications and Networks, page S2352864819301476, Jan. 2020. URL: https://doi.org/10.1016/j.dcan.2019.12.001.

[46] A. Goranović, M. Meisel, L. Fotiadis, S. Wilker, A. Treytl, and T. Sauter. Blockchain applications in microgrids an overview of current projects and concepts. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 6153–6158, Tianjin, China, Nov. 2017. URL: https://doi.org/10.1109/IECON.2017.8217069.

[47] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. page 9, 2018. URL: https://bitcoin.org/bitcoin.pdf.

[48] Coindesk. Coindesk. URL: https://www.coindesk.com/price/bitcoin [last accessed 10.05.2020].

[49] Krishnapriya S and Greeshma Sarath. Securing land registration using blockchain. *Procedia Computer Science*, 171:1708–1715, 2020. URL: https://doi.org/10.1016/j.procs.2020.04.183.

[50] Vitalik Buterin. A next generation smart contract decentralized application platform. page 36, 2013. URL: https://ethereum.org/en/whitepaper/#ethereum-whitepaper.

[51] Stefano Bistarelli, Gianmarco Mazzante, Matteo Micheletti, Leonardo Mostarda, Davide Sestili, and Francesco Tiezzi. Ethereum smart contracts: Analysis and statistics of their source code and opcodes. *Internet of Things*, 11:100198, Sep. 2020. URL: https://doi.org/10.1016/j.iot.2020.100198.

[52] Coindesk. Coindesk. URL: https://www.coindesk.com/price/ethereum [last accessed 10.05.2020].

[53] Helen Eenmaa-Dimitrieva and Maria José Schmidt-Kessen. Creating markets in no-trust environments: The law and economics of smart contracts. *Computer Law Security Review*, 35(1):69–88, Feb. 2019. URL: https://doi.org/10.1016/j.clsr.2018.09.003.

[54] Damiano Di Francesco Maesa and Paolo Mori. Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*, 138:99–114, April 2020. URL: https://doi.org/10.1016/j.jpdc.2019.12.019.

[55] Markos Zachariadis, Garrick Hileman, and Susan V. Scott. Governance and control in distributed ledgers: Understanding the challenges facing blockchain technology in financial services. *Information and Organization*, 29(2):105–117, June 2019. URL: https://doi.org/10.1016/j.infoandorg.2019.03.001.

[56] A. Hrga, M. Gržanić, N. Zhang, and T. Capuder. Decentralized platform for investments and operation of energy communities. In *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, pages 1926–1931, Beijing, China, Nov. 2019. URL: https://doi.org/10.1109/iSPEC48194.2019.8975165.

[57] Tom Lyons. Blockchain innovation in europe. *Report of the European Union Blockchain Observatory and Forum*, July 2018. URL: https://www.eublockchainforum.eu/sites/default/files/reports/20180727_report_innovation_in_europe_light.pdf.

[58] Primavera De Filippi, Morshed Mannan, and Wessel Reijers. Blockchain as a confidence machine: The problem of trust challenges of governance. *Technology in Society*, 62:101284, Aug. 2020. URL: https://doi.org/10.1016/j.techsoc.2020.101284.

[59] Bernd Teufel, Anton Sentic, and Mathias Barmet. Blockchain energy: Blockchain in future energy systems. *Journal of Electronic Science and Technology*, 17(4):100011, Dec. 2019. URL: https://doi.org/10.1016/j.jnlest.2020.100011.

[60] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Differential privacy in blockchain technology: A futuristic approach. *Journal of Parallel and Distributed Computing*, 145:50–74, Nov. 2020. URL: https://doi.org/10.1016/j.jpdc.2020.06.003.

[61] Giuseppe Antonio Pierro and Henrique Rocha. The influence factors on ethereum transaction fees. In IEEE, editor, *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 24–31, May 2019. URL: https://doi.org/10.1109/WETSEB.2019.00010.

[62] Renlord Yang, Toby Murray, Paul Rimba, and Udaya Parampalli. Empirically analyzing ethereum's gas mechanism. In IEEE, editor, *2019 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, pages 310–319, June 2019. URL: https://arxiv.org/pdf/1905.00553.pdf.

[63] Lea Diestelmeier. Changing power: Shifting the role of electricity consumers with blockchain technology – policy implications for eu electricity law. *Energy Policy*, 128:189–196, May 2019. URL: https://doi.org/10.1016/j.enpol.2018.12.065.

[64] M.C. (Annemarie) Buth, A.J. (Anna) Wieczorek, and G.P.J. (Geert) Verbong. The promise of peer-to-peer trading? the potential impact of blockchain on the actor configuration in the dutch electricity system. *Energy Research Social Science*, 53:194–205, July 2019. URL: https://doi.org/10.1016/j.erss.2019.02.021.

[65] Sana Noor, Wentao Yang, Miao Guo, Koen H. van Dam, and Xiaonan Wang. Energy demand side management within micro-grid networks enhanced by blockchain. *Applied Energy*, 228:1385–1398, Oct. 2018. URL: https://doi.org/10.1016/j.apenergy.2018.07.012.

[66] J. A. F. Castellanos, D. Coll-Mayor, and J. A. Notholt. Cryptocurrency as guarantees of origin: Simulating a green certificate market with the ethereum blockchain. In IEEE, editor, *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pages 367–372, July 2017. URL: https://doi.org/10.1109/SEGE.2017.8052827.

[67] Remix. Remix. URL: https://remix.ethereum.org/ [last accessed 06.07.2020].

[68] Metamask. Metamask. URL: https://metamask.io/ [last accessed 06.07.2020].

[69] Sublime Text. Sublime text. URL: https://www.sublimetext.com/ [last accessed 06.07.2020].

[70] Truffle Suite. Truffle suite. URL: https://www.trufflesuite.com/ [last accessed 06.07.2020].