Universidade de Aveiro
2023

Eduardo Afonso de Pinho Soares de Almeida

Algoritmos para Transações de Eletricidade em Mercados Locais de Comunidades de Energia Renovável

Algorithms for Electricity Transactions in Local Renewable Energy Community Markets

Universidade de Aveiro
2023

Eduardo Afonso de Pinho Soares de Almeida

Algoritmos para Transações de Eletricidade em Mercados Locais de Comunidades de Energia Renovável

Algorithms for Electricity Transactions in Local Renewable Energy Community Markets

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Paulo Jorge de Campos Bartolomeu, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor João Miguel Pereira de Almeida, Investigador do Instituto de Telecomunicações.

Este trabalho é financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), através do Programa Operacional Regional de Lisboa (POR LISBOA 2020) e do Programa Operacional Regional do Centro (CENTRO 2020) do Portugal 2020 [Projeto COMSOLVE com o nº 047019 (CENTRO-01-0247-FEDER-047019)].

Este trabalho é financiado pela FCT/MCTES através de fundos nacionais e quando aplicável cofinanciado por fundos comunitários no âmbito do projeto UIDB/50008/2020-UIDP/50008/2020.

Dedico este trabalho a toda a minha família e amigos, em especial aos meus pais e irmã que me apoiaram incansavelmente durante todo o processo.

o júri / the jury

presidente / president	Prof. Doutor Luís Filipe de Seabra Lopes Professor Associado da Universidade de Aveiro
vogais / examiners committee	Doutor Pedro Nuno da Silva Faria Investigador Auxiliar do Instituto Superior de Engenharia do Porto
	Prof. Doutor Paulo Jorge de Campos Bartolomeu Professor Auxiliar em Regime Laboral da Universidade de Aveiro

agradecimentos / acknowledgements

Gostaria de agradecer aos meus orientadores, o professor Paulo Bartolomeu e o investigador João Almeida, por toda a disponibilidade, apoio e incentivos durante a realização desta dissertação.
Gostaria também de agradecer a todos os colegas investigadores que estiveram envolvidos no projecto onde esta dissertação se insere, em especial ao Daniel Andrade e Francisco Monteiro que estiveram envolvidos em todo o projecto desde o início.

Palavras Chave

Mercados de Energia, Prosumidores, Comunidade de Energia Renovável, Algoritmo de Negociação.

Resumo

Com a transformação do mercado da energia de uma estrutura hierárquica descendente para um modelo descentralizado e mais centrado no consumidor, as pessoas passaram a poder produzir, armazenar e vender a sua própria energia, tornando-se prosumidores no ecossistema local de eletricidade. Isto deu origem ao conceito de comunidade de energia, que consiste num grupo de cidadãos ou organizações que trabalham em conjunto para produzir, vender e distribuir energia de fontes renováveis entre os membros da comunidade. Para facilitar a criação destas comunidades, é necessário criar soluções de gestão para as mesmas, pelo que esta dissertação desenvolve um sistema de mercado para estas comunidades de energia renovável. É desenvolvido um sistema onde as medições em tempo real dos contadores de eletricidade podem ser recolhidas e processadas utilizando um algoritmo de negociação de energia. Este sistema pode ser utilizado para testar diferentes tipos de algoritmos e para adaptar o sistema à comunidade em que se insere.

Keywords

Energy Markets, Prosumers, Renewable Energy Community, Trading Algorithm.

Abstract

With the transformation of the energy market from a top-down, hierarchical structure to a decentralized, more consumer-centric model, people are now able to produce, store and sell their own energy, becoming prosumers in the local electricity ecosystem. This has given rise to the concept of an energy community, which is a group of citizens or organizations that work together to produce, sell and distribute energy from renewable sources among the members of the community. To facilitate the creation of these communities, it is necessary to create management solutions for them, so this thesis develops a market system for these renewable energy communities. It develops a system where real-time measurements from electricity meters can be collected and processed using a energy trading algorithm. This system can be used to test different types of algorithms and to adapt the system to the community in which it operates.

Contents

Contents i

List of Figures iii

List of Tables v

Glossário vii

1 Introduction 1

1.1 Context . . . 1

1.2 Motivation . . . 1

1.3 Objectives . . . 2

1.4 Document Organization . . . 2

2 Fundamental Concepts and State-of-the-Art 5

2.1 Energy Trading Systems . . . 5

2.1.1 Bidding Strategies . . . 6

2.1.2 Market Clearing Approaches . . . 8

2.2 Renewable Energy Communities . . . 10

2.2.1 Laws and Directives . . . 11

2.2.2 Stakeholders . . . 11

2.2.3 Control Architectures . . . 11

2.2.4 Community Market Designs . . . 15

2.2.5 Energy Trading Algorithms in RECs . . . 17

2.3 Summary . . . 20

3 Architecture 21

3.1 REC Management Platform . . . 21

3.1.1 REC Architecture . . . 21

3.2 Market Service . . . 23

3.2.1 System Requirements . . . 23

3.2.2 Market Service Architecture . . . 26

4 Implementation **29**

4.1 Market Service Technologies . . . 29

4.2 Smart Meter . . . 30

4.2.1 Low voltage systems . . . 31

4.2.2 Medium voltage and Special low voltage . . . 33

4.3 Meters Microservice . . . 33

4.4 Market Microservice . . . 35

4.4.1 Energy Trading Algorithms . . . 35

4.4.2 Market API . . . 41

5 Results **45**

5.1 Market results for electricity dataset . . . 45

5.2 Market operation using real-time data . . . 50

6 Conclusion **53**

6.1 Conclusions . . . 53

6.2 Future Work . . . 54

References **57**

List of Figures

2.1 Paradigm shift from centralized to distributed energy systems (adapted from [1]). 6
2.2 Energy trading strategies (based on [4]). 7
2.3 Centralized control architecture. 12
2.4 Decentralized control architecture. 13
2.5 Hierarchical control architecture (based on [19]). 15
2.6 Community market design options. 16
2.7 Uniform Price rule (adapted from [4]). 18
2.8 a) Pay Buyers Price; b) Pay Sellers Price (adapted from [4]). 18
2.9 Single-Sided Auction (adapted from [24]). 19

3.1 COMunidade de energia SOLar com integração de Veículos Elétricos (COMSOLVE) architecture. 22
3.2 Market service use cases. 24
3.3 Market service architecture. 27

4.1 Market service technologies. 30
4.2 Smart meter installed in a low voltage system. 32
4.3 Smart meter installed in a medium voltage system. 33
4.4 InfluxDB electricity measurements' data structure. 34
4.5 Possible scenarios for equilibrium point calculation in market algorithms. 37
4.6 Uniform price market algorithm implementation. 38
4.7 Buyers price market algorithm implementation. 39
4.8 Sellers price market algorithm implementation. 39
4.9 Average price market algorithm implementation. 40

5.1 Electricity source for different market algorithms. 46
5.2 Electricity source by month for the *public_grid* and the two groups of algorithms. 47
5.3 Total number of matches for different market algorithms. 48
5.4 Total monetary values for different market algorithms. 49
5.5 Average electricity prices for different market algorithms. 50

5.6 Example of electricity measurements stored in the meters microservice database. 51
5.7 Example of the matches for the timestamp "2023-10-09T14:15:00Z". 51

List of Tables

2.1 Advantages and disadvantages of the different P2P market design options (based on [23]). 17

3.1 System actors. 23

3.2 Use cases description. 25

3.3 Functional requirements. 26

3.4 Non-functional requirements. 26

4.1 Modbus frame. 31

4.2 Register addresses. 32

Glossário

REC Renewable Energy Community
RES Renewable Energy Sources
CC Central Controller
LC Local Controller
MG Microgrid
MGC Microgrid Communities
P2P peer-to-peer
ICT Information and Communications Technology
FiT Feed-in Tariff
NEM Net Energy Metering
DER Distributed Energy Resource
RTP Real-Time Pricing
ZI Zero Intelligence
AA Adaptive Aggressiveness
ABS Agent-based simulation
KPI Key Performance Indicator
DLT Distributed Ledger Technology
RPC Remote Procedure Call
DSO Distribution System Operator
COMSOLVE COMunidade de energia SOLar com integração de Veículos Elétricos
PV PhotoVoltaic
EV Electric Vehicle
GDPR General Data Protection Regulation

CHAPTER 1

Introduction

This chapter presents the context and motivation of this thesis, as well as its objectives and the structure of the document.

1.1 Context

Energy systems are undergoing a profound transformation, moving from a top-down hierarchical structure to a more consumer-centred decentralized model, which will also produce, store and sell its own energy. The electricity markets will have to adapt to this new reality, accommodating the new network architectures and technologies that will support this paradigm. This transition to a more decentralized energy market raises issues of equity and energy accessibility, where it is crucial to ensure that all consumers have access to the benefits of decentralized energy, such as lower energy costs. This is leading to the creation of a more open and diverse energy market.

As important issues such as climate change evolve, transforming the energy system into a more decentralized, clean and consumer-centric system is no longer a choice, but a necessity. Therefore, it is possible to mitigate the increase in global warming, by reducing the world dependence on fossil fuels and opting for renewable energy systems. In addition to the environmental benefits, there are also economic benefits from investing in new technologies that will pay off in the future and reduce consumers' energy bills.

In this context, the concept of Renewable Energy Communities (RECs) arise, consisting of a group of citizens and/or organizations that work together to produce, sell and distribute energy from renewable sources to the members of the community. Although it has been around for some time, the recent changes in the energy market have made it possible to expand it and offer new ways of using it.

1.2 Motivation

That said, it becomes necessary to develop management solutions for RECs with the integration of electric vehicles and energy storage systems based on second-life batteries.

These communities also include decentralized electricity production from photovoltaic panels and the sharing of energy produced by community members, allowing the development of the energy market paradigm centered on the role of *prosumers* - citizens and entities that are both producers and consumers of electricity. In this community ecosystem, members are able to carry out energy purchase and sale transactions directly among themselves, in a secure and distributed manner. These transactions, that could rely on Distributed Ledger Technologys (DLTs) and peer-to-peer (P2P) trading models, are based on the consumption profiles and preferences (e.g. buy/sell price and energy source) of the various users. The REC management entity also has the role of optimizing the community's energy balance, assessing in real time the price of buying and selling electricity to the public grid and making decisions on storage, buying energy for future consumption or selling the generated surplus.

1.3 Objectives

The work developed in the scope of this thesis is integrated in the framework of COMSOLVE project, which aims to create a REC with the inclusion of solar PhotoVoltaic (PV) generation and Electric Vehicle (EV) charging. As an essential component of this project, the main purpose of this thesis is to develop trading algorithms that facilitate the establishment of energy transactions in local energy markets. To this end, the following objectives were defined in order to attain the desired outcomes:

- A preliminary study of the fundamental concepts associated with RECs must be carried out, as well as a survey on the existing trading algorithms for energy markets;
- Define a general system architecture for the implementation of local energy markets in the scope of COMSOLVE project;
- Development of a communications system for obtaining real-time electricity production and consumption measurements of REC buildings.
- Create a storage system for data warehousing of the measured values and the price profiles of REC members;
- Implementation of a market system for testing and validation of different types of trading algorithms;
- Results analysis and comparison of a variety of local market algorithms.

1.4 Document Organization

In addition to this first introductory chapter, this thesis is divided into the following chapters:

- **Chapter 2 - Fundamental Concepts and State-of-the-Art**: this chapter offers an overview of fundamental ideas in the realm of RECs and the corresponding energy trading systems.
- **Chapter 3 - Architecture**: this chapter presents the overall project of this thesis, illustrating its architecture and explaining the services it contains.

- **Chapter 4 - Implementation**: this chapter describes the implementation of the entire system, from obtaining real time readings to describing all the implemented matching algorithms.
- **Chapter 5 - Results**: this chapter presents the results obtained by the system in order to compare the implemented algorithms.
- **Chapter 6 - Conclusion**: This chapter summarizes all the work carried out, as well as a final analysis of the results obtained. It also presents some points for possible future work within the scope of this project.

CHAPTER 2

Fundamental Concepts and State-of-the-Art

This chapter provides an introduction to the essential concepts in the field of RECs and the associated energy trading systems. It begins by presenting energy exchange strategies in various types of markets, then introduces the concept of an energy community and how it can be structured. Finally, some energy market algorithms used in RECs are presented.

2.1 Energy Trading Systems

Energy systems are undergoing a profound transformation, moving from a top-down hierarchical structure to a more decentralized model centered on the consumers, who may also produce, store and sell their own energy. Electricity markets will have to adapt to this new reality, accommodating the new network architectures and technologies that will support this paradigm. Figure 2.1 illustrates this transition from a more centralized model (left) to systems based on the concept of community and centered on the role of the prosumer (right).

With the evolution of the electricity system in Europe, various energy markets have emerged that allow energy to be exchanged across borders in an efficient and sustainable way. The aim is to achieve a single European energy market. In Portugal's case, it is part of the Iberian energy market, which operates in three ways: day-ahead market, intraday market and balancing market. In the day-ahead market, energy is bought and sold for the following day. In the intraday market, energy is exchanged during the day, allowing adjustments to be made according to demand and supply. Finally, the balancing market works in real-time to keep the grid stable. The Iberian market is connected to other markets (France and Morocco), which allows energy to be exchanged with other parts of Europe and beyond.

From the point of view of Distributed Energy Resources (DERs), nowadays there are two main policies: Net Energy Metering (NEM) and Feed-in Tariff (FiT). The FiT appeared to encourage investment and development of renewable energy projects, based on three main points: access to the grid, a long-term contract, and a payment method with fixed selling

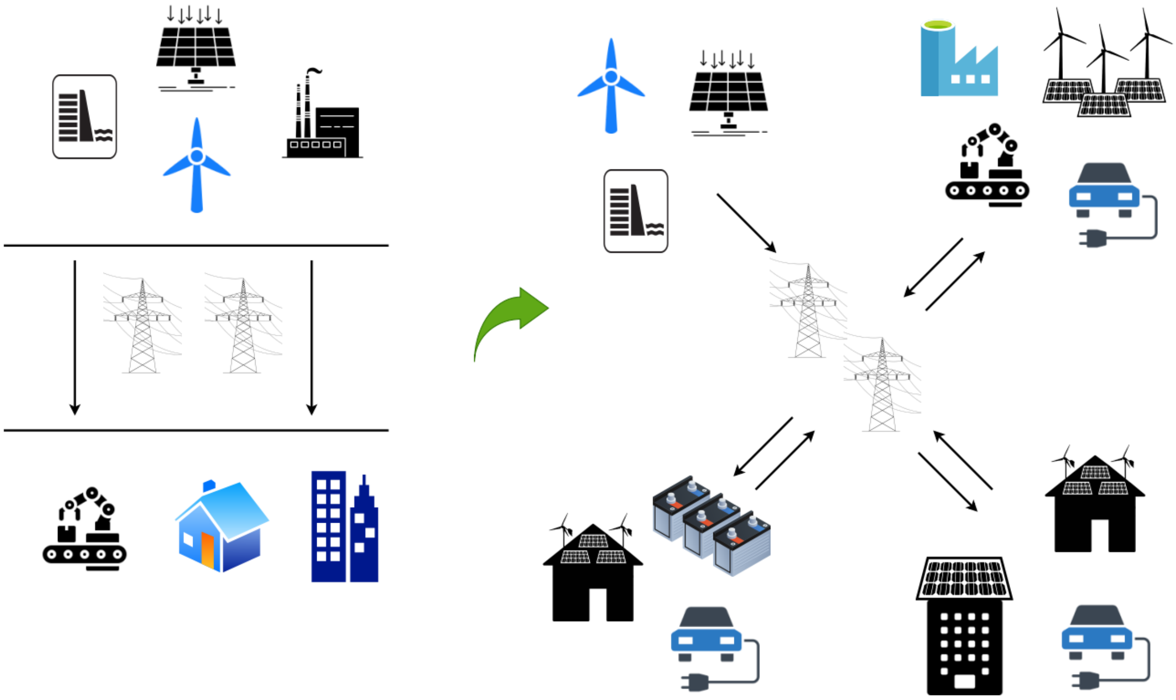

Figure 2.1: Paradigm shift from centralized to distributed energy systems (adapted from [1]).

price based on the cost of the renewable energy source [2]. In the case of NEM it works in a different way, where the energy produced is used to deduct the energy consumed, and the deduction can be done in the month of its production or in a future month where the energy produced does not supply the needs of the consumed energy [3].

In a decentralized energy system, where the focus is on the prosumers, the energy exchange system is divided into two main parts: the first is the bidding strategy, which consists of submitting bids, and the second is the market clearing approach, where the previously obtained bids are used to find a balance point for the purchase and sale of energy. Figure 2.2 illustrates the various parts of the energy exchange system, which will be further elaborated.

2.1.1 Bidding Strategies

Bidding strategies in energy markets consist of producers, consumers, and other market participants submitting bids and offers. These strategies are designed to reduce the risks in the public power grid and maximize profits. These strategies are essential to the functioning of the markets because they influence the price and quantity of energy to be traded. The choice of strategy for the market has to be considered case by case since each market has its own characteristics regarding regulations, objectives, participants (public and private entities) and the market conditions themselves.

There are several bidding strategies, the main ones are:

1. Real-Time Pricing (RTP): it is a strategy where market participants submit their bids in real-time, allowing the market to respond more adequately to changes in the quantities of energy available. The price is defined according to the bids present at that moment, through the intersection of the demand and supply curves, allowing for greater

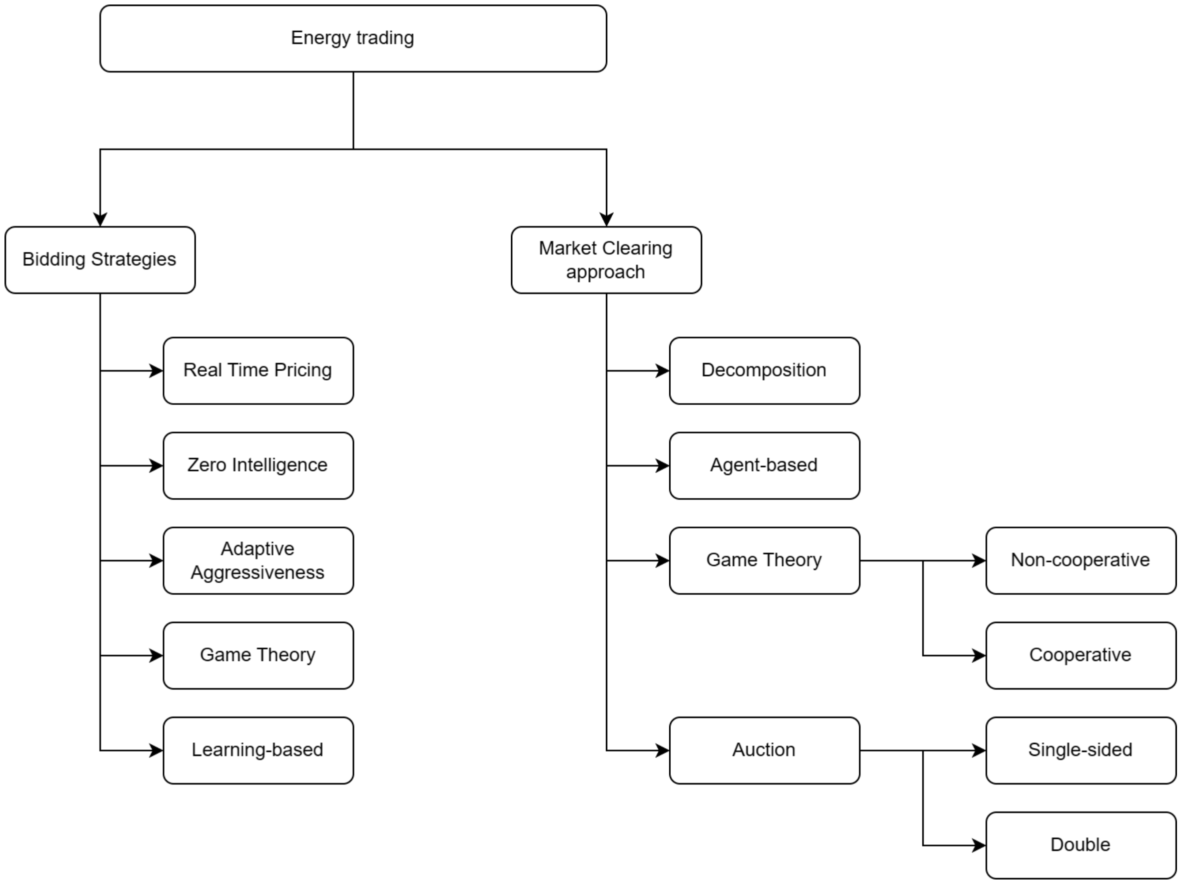

Figure 2.2: Energy trading strategies (based on [4]).

market transparency. In this way the market can become more flexible, as market participants can adjust their bids to the current market situation, leading the market to be more efficient in the distribution of energy. This strategy can be combined with other strategies to build a better market [5].

2. Zero Intelligence (ZI): in this strategy, market participants simply make their bids without using any reasoning to make the bid, nor taking into account the state of the market. This is a strategy commonly used to test markets in simulated environments since it allows the randomized generation of a large amount of data quickly. Although it is a useful strategy, allowing us to verify the efficiency, volatility and reliability of the market, we must take into account that it does not reflect a real market where participants usually have different types of intelligence, the amounts of energy consumed and produced depend on several factors and the possibility of the existence of energy storage batteries [6].
3. Adaptive Aggressiveness (AA): it is a strategy that has as its main objectives the maximization of profits for energy sellers while taking into account the interests of consumers, with the main difference to the previous ones being the adjustment of prices in real-time taking into account market conditions and previous interactions in the market, both by the user in question and by all other users. This strategy provides better market efficiency because it allows adjusting offers taking into account the state of the

market. It also allows greater transparency once price, demand and supply information are shared in real-time. With the information shared by all, it is easier to manage the resources of each one, allowing both energy suppliers and consumers to adjust so that the probability of a power outage is greatly reduced. For this strategy to be successful, energy suppliers need to be able to quickly adapt to changes in the market and even anticipate them [7][8].

4. Game Theory: there are several strategies based on game theory models that allow a mathematical framework to analyze the interactions between each market participant and determine the best strategy. They can be used to determine prices and the behavior of other participants, taking into account the balance between the interest of both energy producers and consumers [9][10]. Some papers compare game theory-based strategies with other strategies, such as the ZI strategy, where they conclude that game theory is the best choice when there is generation at the local level [11].
5. Learning-based: These strategies use machine learning to predict market conditions, the best strategy to use in each situation, and help define the best price for the next bid, based on the energy available in the community by each participant and their previous bids [12]. In some cases, this strategy is only applied in places that have energy storage so that it is possible to predict when to discharge or charge the batteries, and can use different learning models to focus on making a profit or reducing dependence on third-party energy in the case of a local market [13].

2.1.2 Market Clearing Approaches

Various market clearing approaches have been presented over time for the huge variety of markets that exist, with structure, scale, specific rules and participants' behavior as the main parameters to be taken into account. Auction methods are used in local markets, game-theory methods are used to deal with market participants with conflicts, and distributed methods for large-scale markets. For the most part of the markets, a combination of several approaches is needed to achieve a better market [14].

Distributed approaches over the past few years have been in greater demand, being these methods able to reduce market structure costs and provide higher scalability. These methods can be classified into the following categories:

Decomposition approach

The decomposition approach consists of dividing a larger, complex problem into smaller problems with less complexity. With the use of decomposition, it is possible to reduce the costs of a large-scale market, by no longer processing data in a single central point but rather in a distributed manner by each participant. The central point is used to bring together the decisions made at the local level, and then combining them and obtain the global solution.

Agent-based approach

Agent-based simulation (ABS) is a method for modelling and simulating the behavior of energy market players. Each market participant is an agent in this method, with its objectives, strategy and techniques. These agents can be simple to the point of having a single configuration or an intelligent system capable of adapting and learning over time to meet its objectives in the best possible way. This type of approach has several advantages, such as creating a more efficient and well-structured market. Still, on the other hand, there are also several difficulties in its implementation, such as defining the behavior it should have according to different situations, and calibrating real data to create models, not to mention the high computational complexity that this method entails [15].

Game-theory approach

Game theory is a mathematical tool used to analyze the behavior of various participants who make decisions that can lead to conflict or cooperation between them. This provides a way of dealing with smart grid issues according to the behavior of other agents [4], [16]. This approach is divided into two types:

1. Non-cooperative games are used to allow participants to make decisions without any coordination or communication, in the case of participants with partially or totally conflicting interests. This doesn't mean that they can't cooperate, but rather that cooperation, if it exists, doesn't come from communication or coordination between them. Non-cooperative games are further divided into two categories:
    a) static games where players act only once without being aware of other players' decisions.
    b) dynamic games where players act several times and have some information about the previous actions of other players, in which case time plays a very important role in decision-making.
2. Cooperative games are characterized by cooperation between the participants to obtain a greater profit. Initially, terms and conditions are negotiated between those wishing to form a coalition. Coalitions can be classified into 3 types [17]:
    a) canonical coalitional game that analyzes coalition gains and how they are distributed;
    b) coalition formation game that focuses on how players interact to form a coalition;
    c) coalitional graph game that is primarily concerned with establishing communication connectivity among players.

Auction Approach

In energy markets, auction approaches are used to buy and sell energy. From large energy producers to small local markets, this technique is used both in traditional markets and in real-time or futures markets. This technique allows for better efficiency and greater competitiveness within the energy market.

Auctions can be classified according to the number of players, that is, the number of sellers and buyers. When there is only one buyer or seller and several sellers or buyers, they are called single-sided auctions. When there are several buyers and sellers for the same product, these are called double auctions [18].

Single-sided auctions can be divided into four types [4], [18]:

1. Ascending (English Auction): the auction starts with a low price and is incremented by bids made by participants, with the highest bidder winning.
2. Descending (Dutch Auction): the auction starts with a very high price, which is gradually decreased if there are no bidders. As soon as a market participant shows interest in the current value, the auction is closed.
3. First-Price sealed-bid: all market participants submit their bids at the same time, and the bid with the highest value is the winner. In this case, each participant only places one bid, without knowing the bids of the other participants.
4. Second-Price sealed-bid: works in the same way as First-Price sealed-bid where the winner is the participant with the highest bid, but the winner pays only the value of the second highest bid.

Double auctions allow market participants to be buyers at certain times and sellers at other times, and there can be several buyers and sellers at the same time. There are various types of double auctions, including [4]:

1. Sealed-bid double auction: in this auction scenario, all values submitted by market participants are known only to them and the market operator, and it is not possible at any time to know whether other market participants have submitted anything or what they have submitted.
2. Continuous double auction: this is the most common auction method, in which all the values submitted by market participants are organized into two lists, one of bids and one of asks, and these are then sorted in ascending or descending order of price, so that a clearing point is reached at the end, which determines the point of separation between those who are included in the market at that time and those who are excluded from it.
3. Periodical double auction: in this particular case, orders are required to be submitted before a designated time frame known as the market-clearing period. Once this period has elapsed, the market is processed and cleared.
4. Distributed double auction: this is a fully decentralized P2P scenario where any of the participating peers has the ability to process the market. As a result, there is no central entity to manage the entire market.

2.2 Renewable Energy Communities

RECs are based on the generation of energy from Renewable Energy Sources (RES) by groups of citizens or organizations, who jointly produce, sell and distribute this renewable energy across its members. This concept has been created a long time ago, but with the implementation of the new laws and easy access to solar panels by individuals, it is now possible for more ideas to arise and be put into practice. That way, the benefits of the dissemination of RECs are huge and the more people involved, the better. From the reduction of CO_2 emissions to the distribution of cheaper energy, there are a lot of advantages in being part of these communities. With the increasing number of individuals able to take part in the energy system, more businesses and large organizations are being created in order to benefit from energy generation as a group.

2.2.1 Laws and Directives

In order to reduce the emission of greenhouse gases, there has been a major investment in the production of electricity through RES. With this investment, an easier access to electricity with lower carbon footprint can be provided to all consumers. Initially, the European Union (EU) determined that the goal was to reduce the greenhouse gases emissions in at least 40% until 2030 (comparing to 1990) and regarding energy consumption, it was expected to have a 32% contribution from RES. However, these numbers have been updated in the "Fit for 55" package. In this package, the European Commission released legislative proposals regarding the revision of the previous climate and energy legislation for the EU 2030. This release was made under the European Green Deal and it was defined that, until 2030, gas emissions should be reduced by at least 55%, 15 percentage points more than the previous legislation. The new directive also implies that the contribution from RES in energy consumption is expected to grow to at least 42.5%, but aiming for 45%. These values force European countries to take action in the investment of new RES and their infrastructures, in order to obtain a cleaner energy solution.

In Portugal, there is also a plan to achieve carbon neutrality by 2050, with the goal of 80% of electricity production being from renewable sources and the reduction of gas emissions in between 45% and 55% (in comparison to 2005). The laws applied to the microgeneration of electricity took effect in 2007 and was updated in 2014 to the law of self-consumption. Nonetheless, in result of the 2018/2001 Directive of the European Parliament and Council, the self-consumption of renewable electricity will be considered a bigger matter than before. This directive allows the self-consumers to act as a group, allowing the creation of RECs in order to share the electricity produced by their renewable sources.

The decree-law nº162/2019 enables the application of the European directive to the national reality, with the objective of promoting and spreading the decentralized production of electricity, reinforcing the production of energy from RES, and thus reducing the energy dependence of the country and enhancing the social and territorial cohesion. This new regime promotes social inequalities reduction, e.g. by decreasing the electricity bill, the creation of new jobs and better competitiveness of the national companies.

2.2.2 Stakeholders

These communities bring a new agent to the table, the prosumer, who is a member of the community that not only sells its surplus energy to the other members, but also buys energy from them. It is not the only agent in the market, because there can also exist consumers who only buy renewable energy and do not produce it. This allows more people to invest in RESs, as they not only reduce their energy consumption, but also have the incentive of selling or sharing their surplus energy.

2.2.3 Control Architectures

Control architectures play a key role in the efficient management of complex systems, determining how different components interact and cooperate to achieve desired outcomes.

In the field of energy management, especially within Microgrid (MG) systems and hybrid energy configurations, control architectures are critical to optimize resource utilization, ensure stability and respond to changing conditions. There are three basic control architectures [19]: centralized, decentralized and hierarchical.

Centralized Control

For centralized architectures, the focus is on the Central Controller (CC) that optimizes and manages all the system's entities through a communication infrastructure. Each entity uses a Local Controller (LC) to send information to the CC. The controller collects and analyzes all the entities' data, such as energy consumption patterns, power generation and weather conditions. Then, the CC makes the optimal energy system's control and sends its decisions to all LCs, in order to schedule the MG operation. This method, as can be seen in figure 2.3, allows the system to be more flexible, which results in a better use of the energy resources.

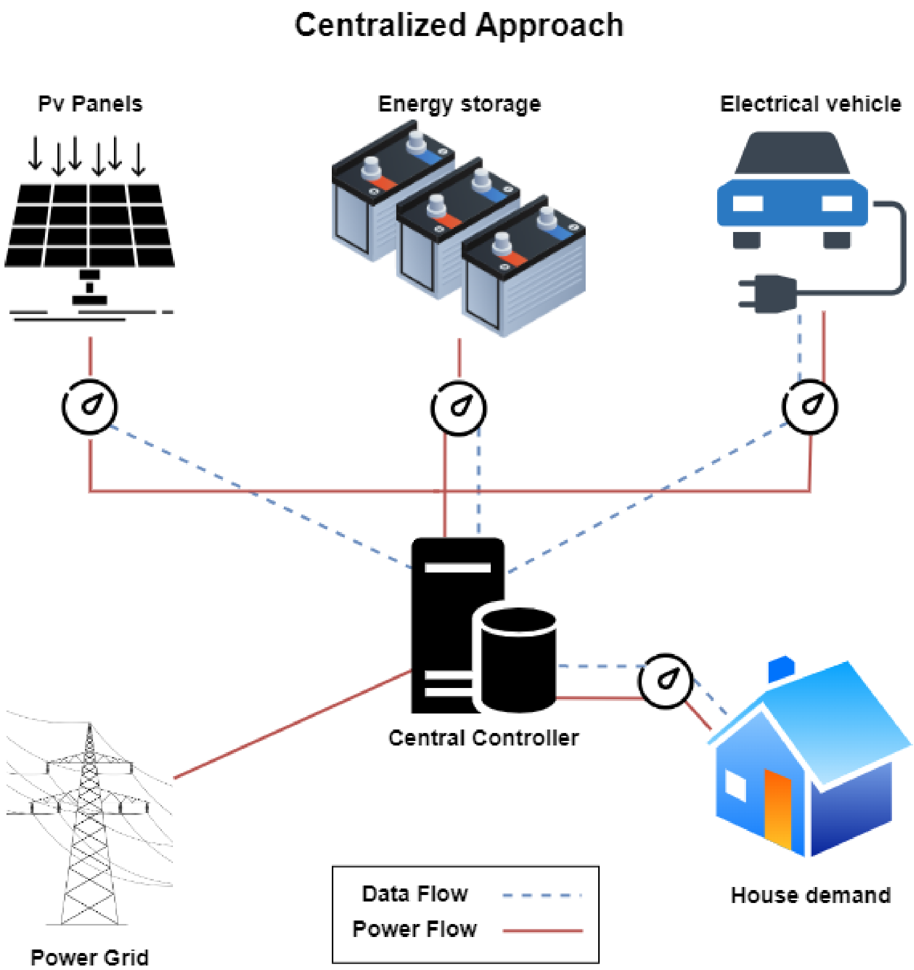

Figure 2.3: Centralized control architecture.

Decentralized Control

Opposing the centralized architectures, in decentralized strategies, each entity is considered autonomous and uses its LC to interact. Generically, the control decisions are made based only on local measurements, which are shared among LCs in a P2P connection. Each LC manages its own energy sources, storage systems and loads without a central controller. However, there are several operation modes that may or may not involve a central entity. The aforementioned method is the fully independent mode, in which the control is resolved without a CC. In a partially independent mode, the LCs communicate among them and with the CC to make a central decision. Lastly, in a fully dependent operation, the LCs generate local decisions and communicate with each other, using the CC as an intermediary. Despite the pliability of these methods, the decentralized strategies underperform in comparison to the centralized control, due to the lack of information about the global MG system and the long response times. In the figure below, a fully independent mode is represented, in which the LCs directly communicate with each other.

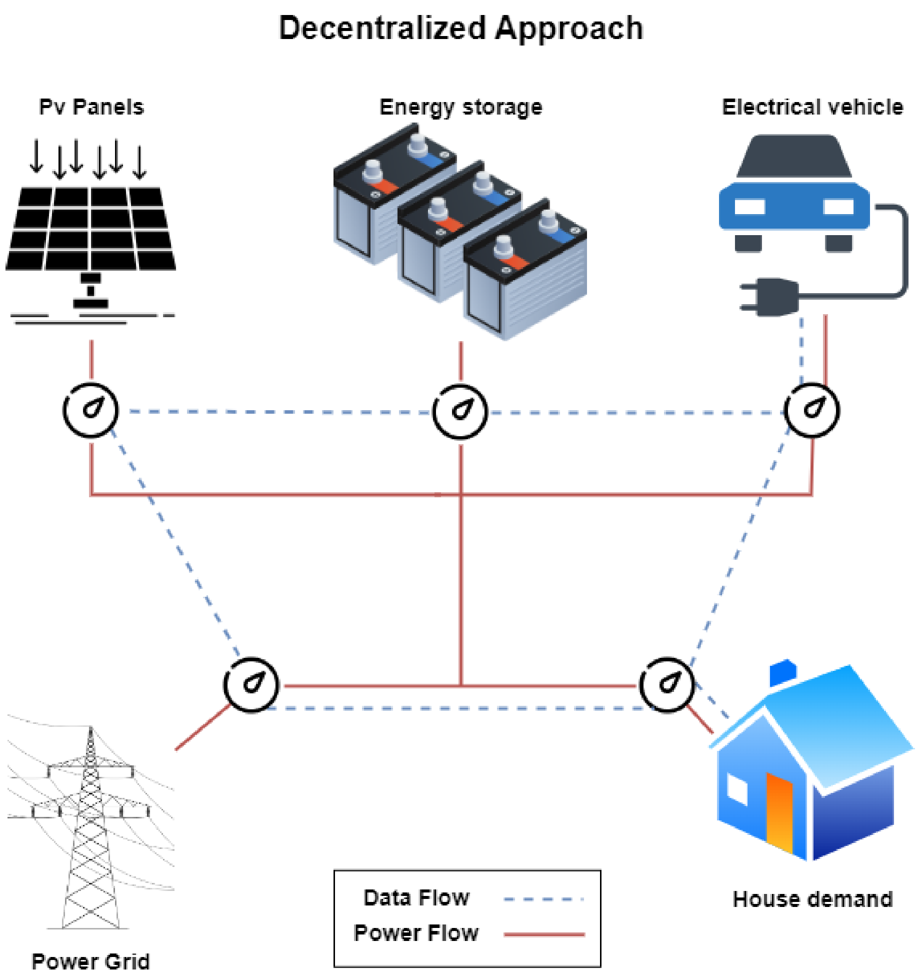

Figure 2.4: Decentralized control architecture.

Hierarchical Control

Since the systems explained above are usually widespread in long distances and need to communicate at a high rate, it is difficult for centralized approaches to succeed, due to the high cost of infrastructure and, at the same time, the high level of coupling and coordination between LCs, which are hard to achieve in decentralized control structures, making the hierarchical approach a viable alternative. This method is primarily preferred in smart grid systems and offers a good strategy to compensate for the other methods' flaws. As referred in [19]–[21], the hierarchical control strategy is based on three levels of control, as shown in figure 2.5.

Firstly, the primary control serves as the voltage and frequency regulator of the local power converters, in order to ensure that energy extraction is being maximized and that energy sharing is available from different sources. Also, it provides local supervision that can detect whether it is better for the MG to operate in grid-connected or island mode, in which the secondary level can act according to this information. Communication at this layer is almost none, because the control is mostly based on local measurements.

Secondly, the secondary control level has the job of synchronizing the MG system and the main power grid, in order to improve the power quality and assure adequate power flow control, by using power dispatching. This is the intermediary between the local system (primary level) and the electrical network (tertiary level) and its job is to manage the DR loads of the LCs and to correct any mismatch between the other layers. That way, this layer uses the information of the third level to send power optimization references to the MG, making the latter more efficient economically and more reliable.

Finally, the tertiary control functions similarly to a Community Operator, controlling the power references of each Distributed Generator to guarantee the optimal operation in both modes and supervising the MG, while considering weather and economical forecast data to satisfy power balance between load consumption and power generation. This level performs optimal economic power dispatching by using powerful optimization algorithms, based on the electricity market price and energy forecasting, resulting in a maximization of the demand-side management process.

These three levels have different response times on their communication to the system. The tertiary operates in an hourly or daily basis, while the secondary intervenes in minutes and the primary responds almost instantaneously.

In order to test the viability of this approach, in [22] there is a proposal for a hierarchical control architecture based on an EMS with an optimization method for Microgrid Communities (MGC). When an MG works on an island mode, i.e., disconnected from the main grid, decentralized control is a possibility, however, it cannot achieve the maximum benefit for the economy of the system because it is isolated from the remaining grid. Considering this, the proposal involves a centralized control system, with a personalized communication system, capable of connecting the MGs over a long distance. For most of the time, the MG will be working in interconnected mode, in which the grid will be stabilizing the electricity of the MGC, by injecting it, when necessary, but when there is a problem with the distribution grid and the MGC works as an island, it will not have enough energy production to hold the community's consumption. So, in this case, the energy storage systems, mostly batteries, will serve the MGC as a backup, in order to restore power quality and stabilize the electricity supply.

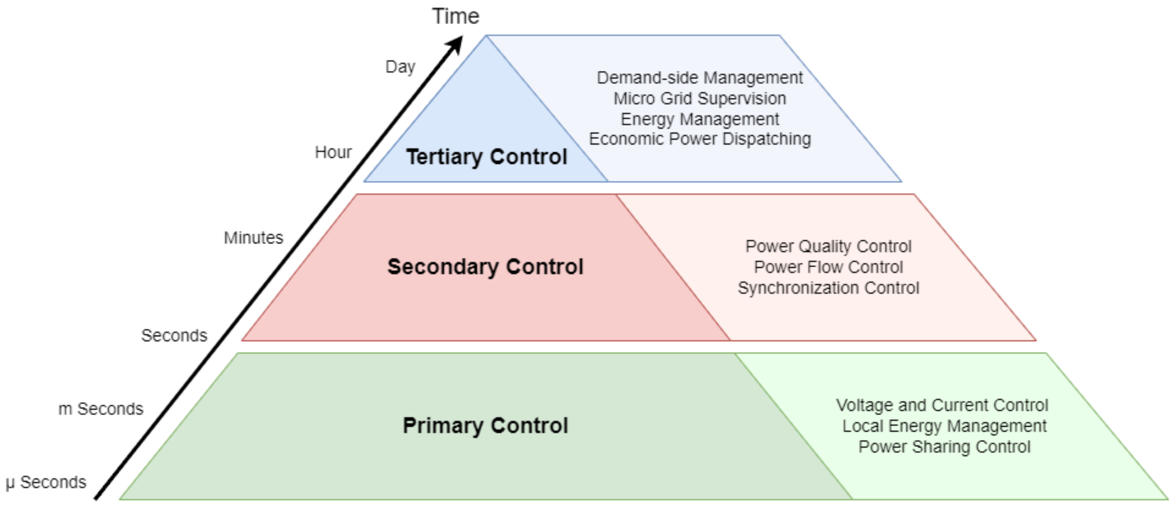

Figure 2.5: Hierarchical control architecture (based on [19]).

2.2.4 Community Market Designs

The consumer centered markets are based on P2P architectures. There are three market designs in consideration [23]: the full P2P, which is completely decentralized; the community-based, that organizes groups of people who share the available resources and the hybrid P2P, which is a mix of the other two.

The full P2P is purely based on energy transactions directly negotiated between peers, without third parties intervention. On the one hand, this model can be very useful for agents, because they are free to set their preferences for the transactions and can negotiate in real time, but on the other hand it could have some scalability problems if the numbers of members increase, leading to a bigger investment in Information and Communications Technology (ICT) infrastructure. Also, since there is no planning on the transactions, it becomes difficult for the grid operators to predict the overall behaviour of the community, which can impact on the safe delivery of energy.

Competing with this model, there is the community-based design, which involves a Community Operator that negotiates the inner community transactions and intermediates with the outer system. This model promotes the social cooperation between the members of the community, in order to pursue a greater energy self-sufficiency. The community manager has a more difficult job on collecting every member's preference in order to satisfy their needs.

Lastly, combining these two designs a third one was created, the hybrid P2P, featuring different aspects of both models. In this design, the peers behave like a community-based market on the bottom level (i.e., the community manager coordinates the transactions), but on the upper level, these peers trade with the energy collectives directly (full P2P). This design has a lot of advantages because it is the most compatible with the current energy model and does not require a lot of extra investment in ICT infrastructure. On the contrary, the community managers have more difficulties, since they have to coordinate the internal and external transactions with other managers and markets.

In figure 2.6, these three different market design options are depicted. The blue circles represent the individual peers in the community as seen in a), describing the full P2P market design, in which peers directly communicate between themselves. For the community-based scenario, the representations can be several, as there are many types of it. In b), the green circles represent the individual peers and the yellow represent the managers of the community. Each black circle forms an energy collective which can work as an island in 1), or as a community who trades with other communities via the community manager in 2). In 3) a decentralized community is represented, where each node can trade with each other, but can trade with other communities as well. This type of community can work as an island too as in 4), where they are self-sufficient and do not require other sources of external energy. In the last representation c), the hybrid P2P is presented, where the two methods are combined and can work as a community or as individual peers.

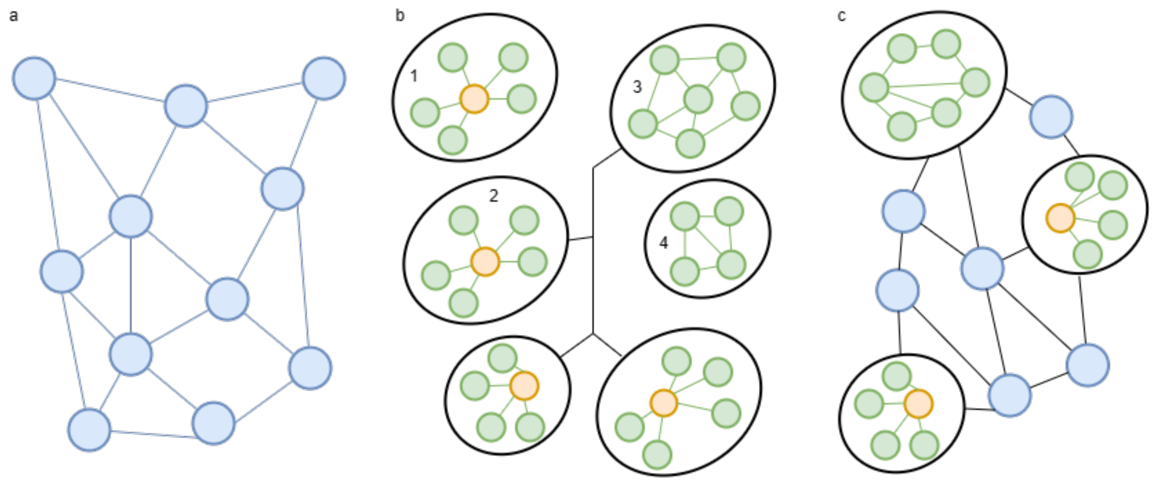

Figure 2.6: Community market design options.

These three methods have their own advantages and disadvantages and have different results when used in certain circumstances. In a review of P2P market designs [23], there was a study of what was the best method out of the three. It was tested on three different communities, involving 19 peers in total. The outcomes of each market type are very specific to the communities in which they were tested, so in different communities the results could be vary significantly. This article also made a revision on the generic pros and cons of the distinct market designs, as seen in table 2.1.

Table 2.1: Advantages and disadvantages of the different P2P market design options (based on [23]).

P2P market design	Main advantages	Main disadvantages
Full P2P market	• Democratization of energy. • Maximum independence and control of individual peers over their trading.	• Needs a bigger investment in ICT infrastructure with more people involved. • No guarantee of safe and high-quality energy delivery. • Difficult to predict by grid operators, due to lack of centralized control.
Community-based market	• Creation of new services by grid operators. • Assembling of a more cooperative community.	• Harder for community managers to reach an agreement between all members' preferences. • Possible unfair or biased delivery of energy.
Hybrid P2P market	• ICT infrastructure is scalable to the whole system. • Most compatible with the current system. • Most predictable to the grid operators.	• Difficult to coordinate the internal trades with the trades between high-level agents.

2.2.5 Energy Trading Algorithms in RECs

Energy communities utilize diverse algorithms for energy exchanges, with auctions being the most commonly employed approach. The double auction is predominant among the various auction types as it facilitates several buyers and sellers to participate in the market with different offers concurrently. Several algorithms that are commonly used in the literature are presented below.

Uniform Price

This algorithm consists of a Sealed-bid Double Auction with a uniform price rule [4], where all the buy and sell offers are submitted. The buy bids are sorted upwards in price and the sell bids are sorted downwards in price. The point of intersection of the organized buy and sell offers represents the equilibrium point. All the bids to the left of the equilibrium point will trade with each other, the rest will be discarded as they are beyond the equilibrium point, as shown in figure 2.7. The price for these energy exchanges corresponds to the price at the equilibrium point.

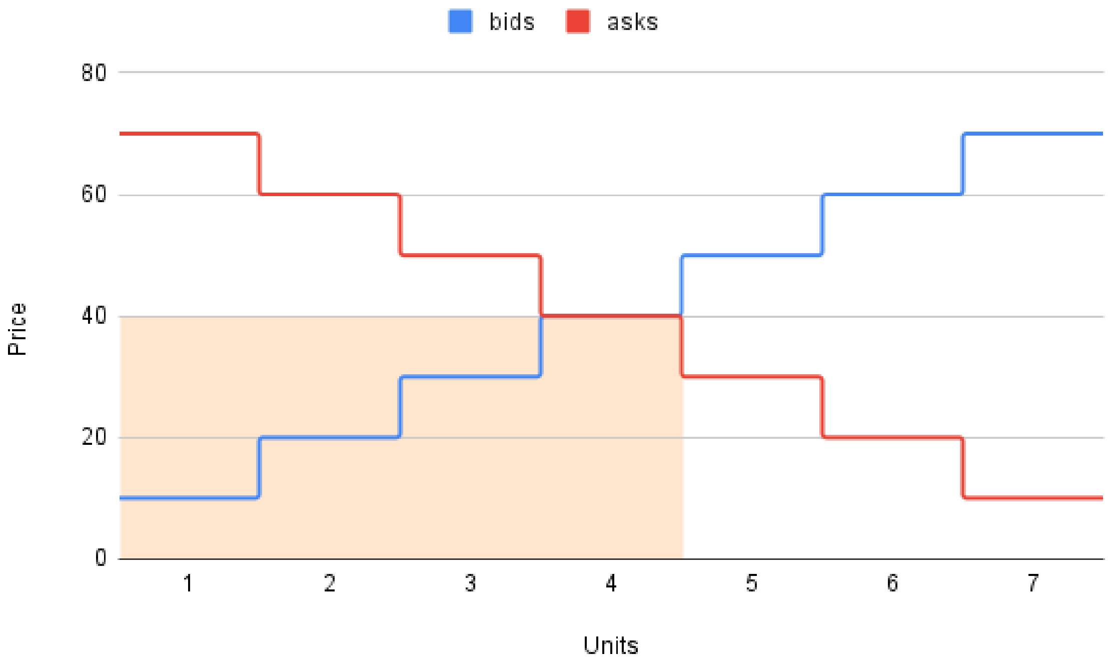

Figure 2.7: Uniform Price rule (adapted from [4]).

Pay Buyers or Sellers Price

These two methods work in the same way as the previous one, with the difference in the Pay Buyers Price case that the price is defined by the purchase orders as shown in Figure 2.8 a). On the other hand, in the Pay Sellers Price case, the price is set by the sell orders as shown in Figure 2.8 b).

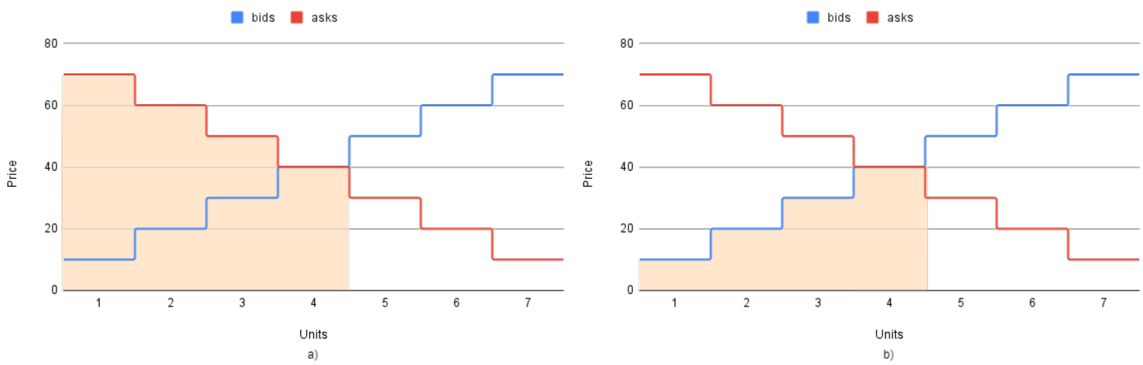

Figure 2.8: a) Pay Buyers Price; b) Pay Sellers Price (adapted from [4]).

Single-Sided Auction

In the Single-Sided Auction [24], the sales orders are organized in ascending order of price, in the same way as the previous ones, while the purchase orders are not organized, only the total energy demand of these orders is taken into account. The energy price is equal to the price of the sales order that fulfills the energy demanded by the purchase orders, as can be seen in figure 2.9.

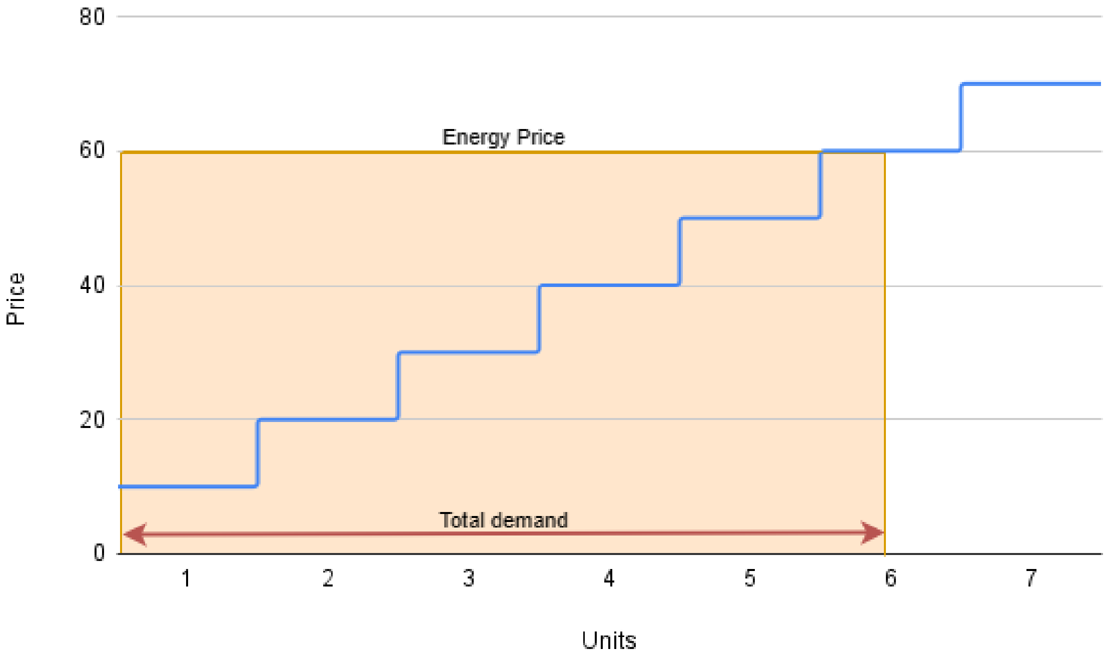

Figure 2.9: Single-Sided Auction (adapted from [24]).

Bill Sharing

This method consists of a community sharing the energy produced within it and each member buying the rest from the public grid [25]. In this case, the energy is distributed free of charge to all members of the community. When there is excess energy within the community, it is sold to the public grid. All orders with excess energy after the exchange within the community will sell the unused surplus to the public grid. On the other hand, if there is a shortage of energy within the community, all requests with a shortage of energy will buy the percentage of the shortage within the community from the public grid. In the exceptional case where the amount of energy produced and consumed within the community is equal, the monetary amount traded is zero.

Mid-Market Rate

In this case, the price of energy within the community is defined by the average between the purchase and sale price to the public grid [25]. As in the previous case (Bill Sharing), the existence of an energy deficit or surplus in the community is verified and dealt with in the same way. Energy, which in the previous case was shared for free within the community, will be exchanged between all members at the price defined by that mid-value.

2.3 Summary

This chapter described the basic concepts related to energy markets, their mode of operation, and how an energy community works in general. Given that the main objective of this thesis is to implement trading algorithms for local energy markets, all the algorithms presented in more detail for the use case of RECs will be deployed. As a result, only auction-based algorithms will be implemented, being the most appropriate for this thesis, at least for this first stage of REC development, although there are other types of algorithms, such as the ones relying on game theory principles. Given the focus of the project in which this thesis is integrated, the bids should be submitted using the Zero Intelligence strategy, since the submission will not take into account the state of the market. Various tests will be carried out on the trading algorithms, making it easier to carry out these tests with this submission strategy. As the main objective is to create a REC that relies on a community manager, the best approach is to use a community with a central control unit, in which the REC manager can have full control over the community.

CHAPTER 3

Architecture

This chapter presents the overall project of this thesis, illustrating its architecture and explaining the services it contains. Next, the actors, their associated use cases and the functional and non-functional requirements of the system are described. Finally, the architecture of the Energy Market Service is presented.

3.1 REC Management Platform

This thesis is part of the COMSOLVE project [26], where the aim is to create a management platform for RECs with the integration of electric vehicles and energy storage systems. The project includes the production of energy from photovoltaic panels and the sharing of energy between members of the community, thereby developing market systems in which the consumer (members of the community) is the main player. In this community, it will be possible for its members to exchange energy directly with each other, with the possibility of buying energy through transactions between them or buying energy from an external source (such as an energy operator). All energy transactions within the community are based on blockchain technology and P2P exchange models, where each member has their profiles and preferences (such as energy purchase and sale prices). There will be a community management entity able to optimize and balance the REC operation, by interacting in real-time with the electricity transactions inside the community and making decisions about energy storage.

3.1.1 REC Architecture

The COMSOLVE project implements a complex architecture, integrating various internal and external services to the REC Management Platform, as illustrated in figure 3.1. The platform is represented within the larger rectangle, where it is divided into two main parts: front-end and back-end. The front-end consists of a web application for community members and an application for community management. The back-end is made up of various services including:

- Back Services: describes all the REC features and data that the users will access through the application.

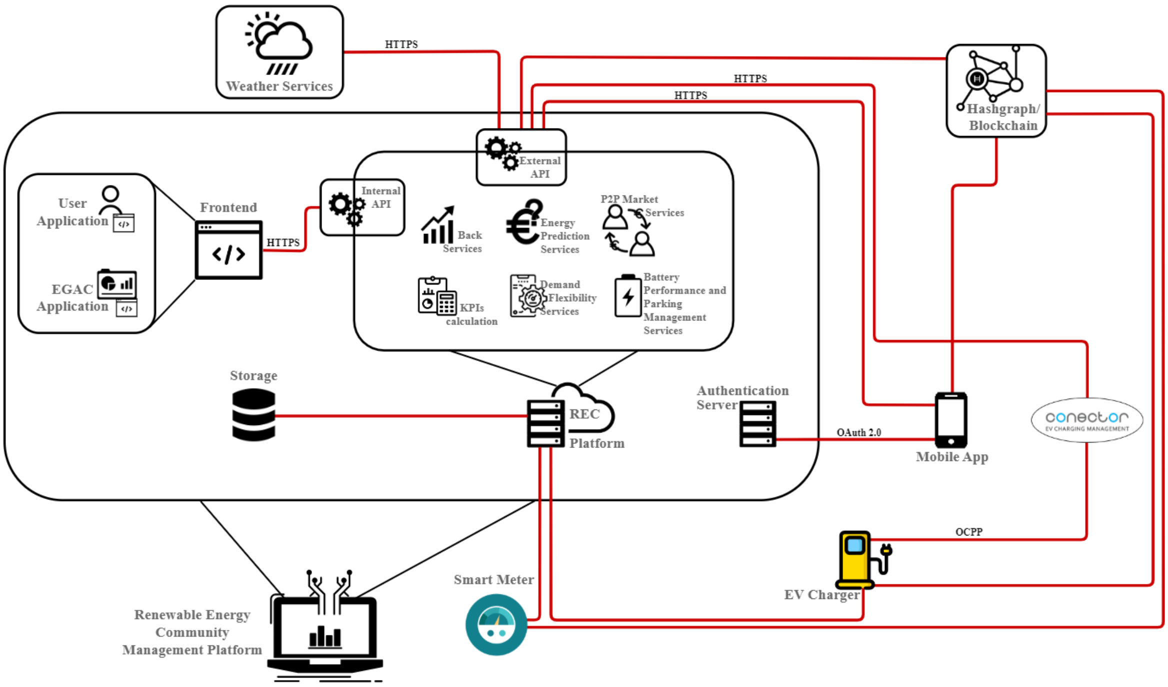

Figure 3.1: COMSOLVE architecture.

- Energy Prediction Services: provides electricity consumption and PV generation forecasts of the REC buildings.
- P2P Market Services: establishes energy transactions among the REC members according to their preferences and chosen market algorithm.
- Key Performance Indicators (KPIs) Calculation: analyses local data to calculate essential variables that assist the REC management process.
- Demand Flexibility Services: uses various methods to optimize and shift energy usage, based on the KPIs Calculation and Energy Prediction Services.
- Battery Performance and Parking Management Services: utilizes battery data to monitor their performance and degradation over time and controls the parking sensors in the EV charging slots to detect illegal occupation of those parking spots.

External to the platform are other important services for the project, such as an authentication server that allows users to be synchronized between the REC platform and the Connector platform. The Connector platform is the one used to manage the EV chargers. Another interconnection present is with the Weather Services used by the internal forecasting system. Last but not least, the DLT Hedera Hashgraph is used for secure transactions between community members.

The scope of this thesis is limited to the development of the smart meters for collecting energy measurements and the implementation of the P2P Market Services in the back-end of the REC Management Platform.

3.2 Market Service

The market service consists of a set of microservices that enable the coordination of the entire market system, from storing smart meters' measurements, processing and creating P2P transactions, creating price profiles for each member of the community and even creating and controlling all the operations in the DLT. This work focuses on the process that encompasses handling the smart meters' data up until computing the P2P matches according to the selected market algorithm. The DLT transactions part and the overall system security are outside the scope of this document.

3.2.1 System Requirements

This section presents the actors, use cases and system requirements for the system architecture.

Actors

This service has three types of actors: the first is the smart meter, which consists of a programmable device located and connected to the electricity meter at the entrance of a building, with the aim of sending real-time measurements to the central server and storing a wallet with which transactions are automatically paid. The second actor is a member of the community who can be either a prosumer or a consumer, where a prosumer is a member of the community who both produces and consumes energy, while a consumer only consumes energy and doesn't produce it, as only the prosumer has the means to produce energy. This actor has the role of defining his price profile, in the case of a consumer only the buying profile, while the consumer has both profiles (buying and selling). The last actor is the manager, who acts as the controlling entity of the REC. Not only does it have full access to the community's data, but it is also responsible for making decisions in order to maintain a balance within the community. Table 3.1 shows all the actors involved in the system, as well as their role in it.

Table 3.1: System actors.

Actor	Role
Smart Meter	The Smart Meter is a programmable device that captures the readings, sends them to the central server and stores the wallet for payment transactions.
Prosumer/Consumer	Prosumer consumes and produces energy from the community. They can define their own price profiles for both buying and selling energy. Consumers, on the other hand, only consume energy and only have to define their energy purchase prices.
Manager	The manager acts as the REC control entity.

Use Cases

The use cases associated with the interactions between the actors and the market service developed in this work are presented in figure 3.2. Additionally, in table 3.2 one can find a description for each use case. The smart meter has two main use cases: one is to send real-time measurements from the electricity meter at the entrance of a building; and the second is related with the authentication process so that the server allows it to send measurements. As for the manager, it also has two main use cases, one consisting of consulting the data from the measurements sent by the smart meters; and a second consisting of visualizing and controlling the prices charged for energy transactions over time. Finally, prosumers and consumers have four use cases: one where they authenticate with the smart meter to carry out certain operations in the service; a second where they define their price profiles for buying and selling in the case of prosumers and buying only in the case of consumers; and two data consumption use cases, one where they visualize the measurements sent by their smart meter and a second where they visualize their energy matches over time.

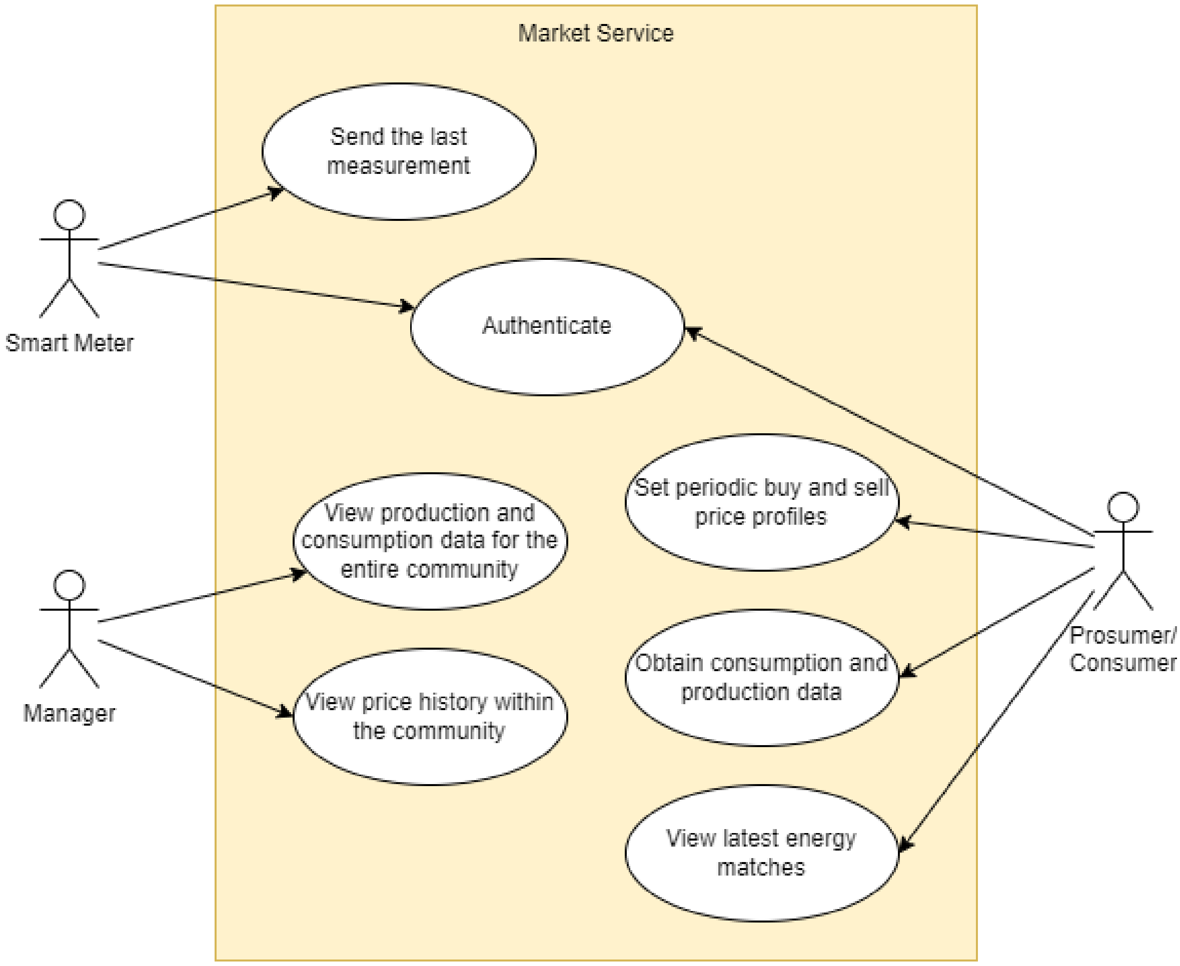

Figure 3.2: Market service use cases.

Table 3.2: Use cases description.

Use Case	Description
Send the last measurement	At each predefined time interval, the smart meter reads and sends the current measurement to the central server.
Authenticate	Use of the authentication system to be able to send or update data in the system.
Set periodic buy and sell price profiles	Each member of the community can define their price profiles for each time interval. Buy and sell prices are two independent profiles.
Obtain consumption and production data	Each member can access their consumption and production data for any previous time interval.
View latest energy matches	Each member can view their latest energy matches, obtaining data on where their energy came from or where it went.
View production and consumption data for the entire community	The managing entity can view all consumption and production data within the community.
View price history within the community	The manager can view the community's price history and matches.

Functional Requirements

Table 3.3 describes the functional requirements of the market service. These relate to the guaranteed transmission of measurements, i.e. a smart meter must be able to resend a reading until it receives confirmation that this reading has been recorded. The market structure must be prepared to change the main algorithm without changing the rest of the structure. The data from the measurements must be processed each period, even if not all smart meters have managed to send their measurements, and these are discarded until the situation is regularised. Community members should be able to change their price profile for the future at any time. Any surplus or shortfall of energy in the community should be recorded as a transaction to or from the public grid, respectively.

Table 3.3: Functional requirements.

ID	Functional Requirements
FR-1	Meters shall be able to resend any measurement to the REC platform if requested by it and no acknowledgment has been received.
FR-2	Meters must store measurements until an acknowledgment has been received from the REC Platform.
FR-3	The market should be able to work with various types of algorithms without the need to make changes to the service structure.
FR-4	In the event of a failure to send readings, the system should still work, discarding the meters that failed to send readings.
FR-5	Energy that is not exchanged within the community should automatically be assigned to the public grid.
FR-6	The price profiles of each member can be changed at any time as long as they are for a future period.

Non-functional Requirements

Table 3.4 describes the non-functional requirements of the market service. The main purpose of these requirements is to ensure that all communications are secure, preventing the monitoring of consumption and production habits by entities other than the system itself, and that the system complies with General Data Protection Regulation (GDPR) standards. The time synchronization between all smart meters must be minimal and never exceed three seconds. Finally, each meter must send its readings for each time period as quickly as possible. In the worst case, it has five minutes to do so, after which the market service will discard it for that time period.

Table 3.4: Non-functional requirements.

ID	Non-functional Requirement	Type
NFR-1	Time synchronization among all meters must be within 3 seconds tolerance.	Environment
NFR-2	Prevent the monitoring of habit consumption and production by other users.	Confidentiality
NFR-3	All the machine-to-machine communications must be encrypted.	Security
NFR-4	Ensure compatibility with the GDPR.	Auditing
NFR-5	Meters must submit a measurement within a maximum of 5 minutes.	Availability

3.2.2 Market Service Architecture

Based on the requirements presented above, it was decided to divide the service into several microservices, resulting in the architecture shown in figure 3.3. The choice of a microservice architecture allows several parts of the service to be developed at the same time without one depending on the state of development of the other microservice, since there can be distinct people involved in the continuous development of the service.

The market service is designed to interact with two types of external devices: the first is the smart meter and the second is any type of external service. The smart meter is the device

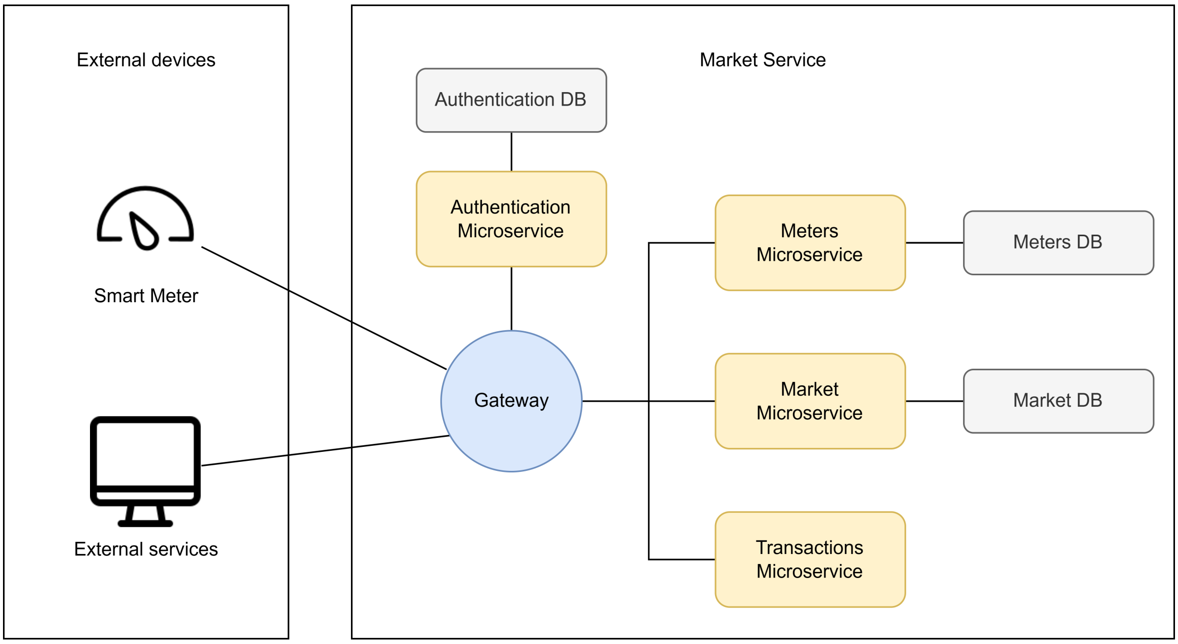

Figure 3.3: Market service architecture.

that is installed next to the electricity meter at the entrance of a building, with the aim of communicating the meter's measurements every 15 minutes and then receiving transactions to pay using a wallet stored in the meter, so that transactions can be paid for without a person having to confirm them on their own device. On the other hand, external services encompass any type of front-end or back-end service that wants to consume or modify data from the service, such as changing a user's pricing profile, adding or removing users, consulting the consumption and production of the community or a specific user, or even just checking the status of internal microservices.

The internal part of the service consists of three main microservices, an authentication microservice and a gateway. The gateway acts as the entry point for any type of external communication. It is responsible for checking that this type of communication is permitted for the external entity that has initiated it, and if authentication is required, it forwards the request to the authentication service. The gateway forwards the request to the microservice that will serve it, only after the authentication process is successfully completed. The meter microservice is where the real-time measurements sent by the meters are stored for later access. The market microservice acts as a computing algorithm, by processing all the electricity measurements from the meters microservice at a predefined rate and employing a market algorithm to create energy matches between community members and allocate the remaining energy to the public grid. Finally, the transactions microservice acts as the point to which transactions to be paid by smart meters are sent, but also as a control point for transactions that have already been made and are being processed. A transaction is not completed until this microservice has verified that everything has been performed correctly. The implementation details regarding the gateway, authentication and transactions microservices are outside the scope of this thesis.

CHAPTER 4

Implementation

This chapter describes the technologies used to implement the whole system. The operation of a smart meter developed for different voltage systems is presented. The meter and market microservice implementations are also described. In the market microservice, in addition to the overall service structure, all the implemented matching algorithms are detailed.

4.1 Market Service Technologies

The market service was implemented in several phases, using a microservices architecture that made it easier to develop several parts of the service at the same time and at different stages of development. The service was also divided into two main parts. The first part consists of taking real-time measurements, storing this data and then processing it to create energy matches. The second part consists of performing DLT transactions and implementing security mechanisms in the system. The aim of this thesis is to implement the first part, while the second is mentioned in some points, but not in great depth, as it is out of the scope of this work.

All the market service modules were implemented using a docker container to ensure that all the modules work on any machine they are installed on. As a means of communication within the service, gRPC was used. This is a high-performance Remote Procedure Call (RPC) framework developed by Google that uses protocol buffers and HTTP/2, which allows communication between distributed services, regardless of the communication language being used. As for the external communication method, HTTPS was used since some of the external services may not support HTTP/2 communication but only HTTP/1.1. As for the framework used, both the gateway and the authentication, meters and transactions microservices use NestJS, which is a NodeJS based framework in which the language used is TypeScript. On the other hand, the market microservice was implemented using Python since it implements various algorithms that benefit from the use of libraries present in it and, for the purpose of comparing algorithms, it allows for a better form of fast data consumption. In addition, the market microservice is made up of a MongoDB database that allows large amounts of data to be inserted and read quickly. In the meters microservice, a time series database was used, in this case InfluxDB, since the measurements data is always used in temporal order. The technologies used in each market service module are shown in Figure 4.1.

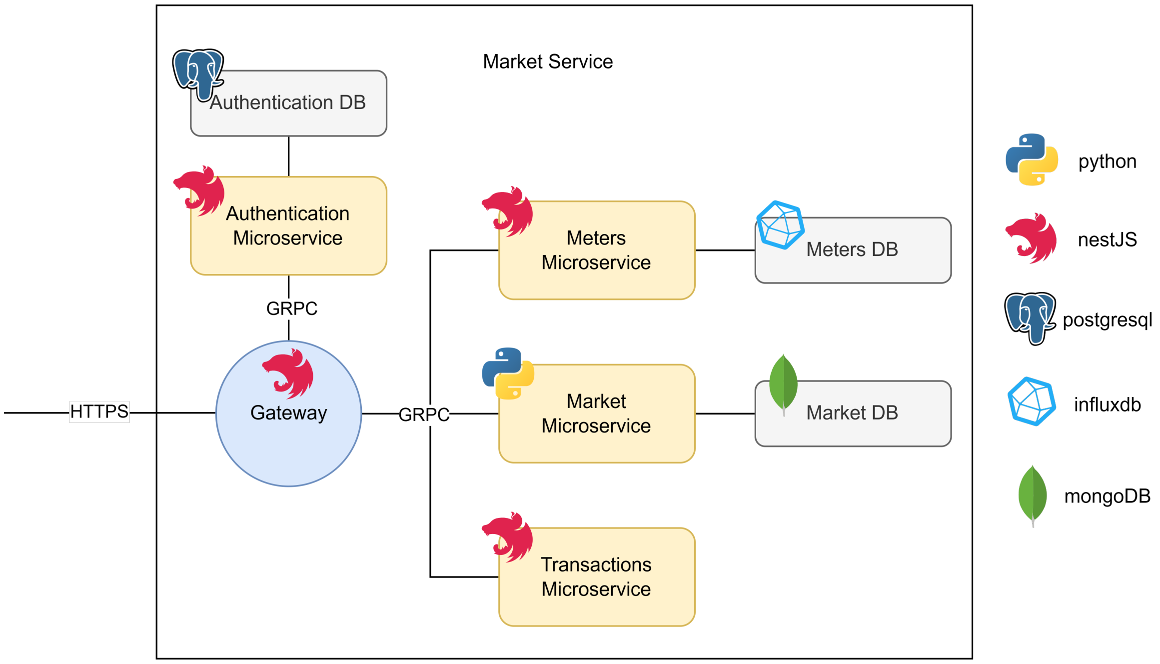

Figure 4.1: Market service technologies.

4.2 Smart Meter

The main purpose of the smart meter is to send electricity measurements in real time every 15-minutes, according to the availability of new data from the official electricity meter installed by the Distribution System Operator (DSO). The device is also used to pay for transactions and perform various tests. A Raspberry Pi 4 was chosen to achieve these goals, as it is a versatile device and can perform different types of tests for other parts of the project. A Sixfab 5G Development Kit for Raspberry Pi was used as the communication module, allowing the device to communicate easily wherever it is installed, even if it does not have access to a wired or Wi-Fi network.

Raspberry Pi OS Lite (64-bit) was used as the operating system because it is the system developed by the brand of the device and it is possible to implement all the modules intended to use. The Lite version was chosen because the device does not require a graphical interface, as all interactions with it will be remote or local via an SSH connection. To facilitate local communication with the device, an invisible Wi-Fi network was set up, which only allows users who have *a priori* knowledge of its existence to connect to it, so that it is not necessary to use cables to interact with the device locally, communication via cable will only be necessary in extreme cases of maintenance and debugging. For remote communication, an OpenVPN client was added to the device so that it can be accessed remotely from anywhere. This way, there is no need to travel to the site to carry out a new configuration.

The real-time metering devices were originally designed for low voltage systems, but as the project developed, it became necessary to implement devices also for special low and medium voltage systems. This is due to the fact that for obtaining these measurements from electricity meters at the entrance of buildings, these higher voltage systems do not use the same standards to obtain the electricity measurements.

4.2.1 Low voltage systems

In the case of low voltage systems, the electronic meter is accessed via a HAN port. In order to access this port, a request must be made on the E-REDES DSO website or to the retail supplier itself so that a technician can come to the meter's location and provide access to the port. In order to make the request, certain conditions must be met: the meter to which the request is made must be smart, the installation must be low voltage and the equipment to be installed must meet the requirements specified in [27]. In the case of the devices installed in this work, there was no problem in meeting these requirements as the equipment is powered externally to the meter and only uses the HAN port to request and receive readings.

To access the port, a cable is used with an RJ12 connector at one end, which is connected to the HAN port, and an RS485 to USB converter at the other end, so that it can be connected to the Raspberry Pi. The Modbus communication protocol is used to communicate via the HAN port and must have the structure shown in table 4.1.

Table 4.1: Modbus frame.

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 to 252 bytes	2 bytes

Several registers can be accessed through the HAN port, but for this project only the four shown in table 4.2 have been selected. All the registers and the way to communicate through the port are described in document DEF-C44-509/N [28], from which the Modbus frame is obtained for each desired register. The slave address for all the pretended registers is "0x01". The function code is "0x04", since one wants to perform readings. As for the data parameter, this is the combination of the address of each register (from table 4.2) together with the desired number of registers, in this case "0x0001" for all of them. Finally, the CRC is obtained by calculating the checksum of the frame through the CRC-16 Modbus format generated in normal big endian mode.

A Python program was implemented on the Raspberry to take readings every 15 minutes via the HAN port and then send them to the market service, which runs every 15 minutes via a pre-configured *cron job*. To obtain the readings via the HAN port, a serial communication is opened where the frame Modbus of each register is written byte by byte and, immediately afterwards, 9 bytes are read via the same serial communication channel, corresponding to the response of the request sent. Only the data part corresponding to the range from byte 4 to 7 inclusive is utilized from this response. Once the four requests have been made and their values collected, the data is sent to the market service by sending a post request.

Table 4.2: Register addresses.

Index	Address	Description	Type	Unit	Modbus frame
22	0x0016	Active energy import (+A)	Double long unsigned	Wh	0x010400160001D00E
23	0x0017	Active energy export (-A)	Double long unsigned	Wh	0x01040017000181CE
24	0x0018	Reactive energy QI (+RI)	Double long unsigned	VArh	0x010400180001B1CD
27	0x001B	Reactive energy QIV (-RC)	Double long unsigned	VArh	0x0104001B000141CD

The object sent consists of a hash created from the meter ID (device ID), the read timestamp, an API key previously created and entered into the device when it was configured, and finally the device's MAC address. In addition to this hash, which is sent as an header, a JSON consisting of the four collected measurements, the device ID and the reading timestamp is sent. In the event of a failure, three attempts are made by each process both for each read performed over the HAN port and for the transmission to the market service. If something fails after three attempts, the error is recorded in a log file and the reading is not received by the market service in the predefined interval. Figure 4.2 shows an installation of the system in a household with photovoltaic panels in the municipality of Ílhavo, where various tests of the system were carried out.

Figure 4.2: Smart meter installed in a low voltage system.

4.2.2 Medium voltage and Special low voltage

For medium and special low voltage scenarios, the four registers (active energy import, active energy export, reactive energy QI and reactive energy QIV) are obtained using electric signals supplied by the meter. To obtain these signals, a special request must be made to E-REDES DSO, which installs four channels to measure the signals. On the Raspberry Pi, using its GPIOs in INPUT mode and a C program using the pigpio library, the signals obtained from the meter are counted and stored in a shared memory. Using the ctypes library, which provides access to the shared memory, the current value of the signal counter is obtained for each of the registers and then sent to the market service, as in the previous low-voltage case. Figure 4.3 shows the installation of the described system in "Casa da Cultura de Ílhavo" municipal building, a medium voltage installation.

Figure 4.3: Smart meter installed in a medium voltage system.

4.3 Meters Microservice

The meters microservice is the module where all the measurements sent by the meters are stored and later used by other services, both for processing energy exchanges within the community and for use in consumption forecasting services. The measurements stored here are also used for visualization on a dashboard available to each member of the community.

A timeseries InfluxDB database is used to store these measurements. This database is organized into buckets, where an API token is generated to interact with the bucket and define the permissions it has over a given bucket. Each bucket is made up of a mandatory column layout, starting with the "_time" field, in which the timestamp of the reading is stored. The column "_measurement" acts as an identifier of the data type, in this case "energy", "_field" identifies the reading record stored in that row, while "_value" stores the value of the reading. There is also the possibility to create various optional parameters, in this case the optional parameter "device" was created which identifies the meter to which the reading belongs. Figure 4.4 shows the structure of the data stored in the database through a dashboard provided by the database itself.

table mean	_measurement group string	_field group string	_value no group double	_start group dateTime:RFC3339	_stop group dateTime:RFC3339	_time no group dateTime:RFC3339	device group string
0	energy	Active export	0	2023-10-13T15:46:31.807Z	2023-10-13T16:01:31.807Z	2023-10-13T16:00:10.000Z	es-sms-15
1	energy	Active import	4852917	2023-10-13T15:46:31.807Z	2023-10-13T16:01:31.807Z	2023-10-13T16:00:10.000Z	es-sms-15
2	energy	Reactive QI(+Ri)	86989	2023-10-13T15:46:31.807Z	2023-10-13T16:01:31.807Z	2023-10-13T16:00:10.000Z	es-sms-15
3	energy	Reactive QIV(-Rc)	1696699	2023-10-13T15:46:31.807Z	2023-10-13T16:01:31.807Z	2023-10-13T16:00:10.000Z	es-sms-15

Figure 4.4: InfluxDB electricity measurements' data structure.

This microservice provides two points of communication:

- *AddMeasurement()*, which allows a smart meter to send a new measurement as long as it uses the message format described in code 1. This returns to the smart meter an object consisting of a status code, an optional message describing the status code and a list of transactions that the smart meter has not yet paid. These transactions are obtained via a request to the transactions microservice made before sending the response to the meter.

```
message MeterEntry{
    string deviceId = 1;
    int32 activeImport = 2;
    int32 activeExport = 3;
    int32 reactiveInductive = 4;
    int32 reactiveCapacitive = 5;
    string timestamp = 6;
}
```

Code 1: Proto message MeterEntry.

- *RetrieveMeasurement()*, which allows a query to be made on the data stored in the microservice. This query can include a filter of optional parameters. This way, the filter may be composed of: a *startInterval*, which allows the user to set the timestamp that defines from which moment it wants to retrieve the data until the current time; a *deviceId*, which allows the user to retrieve data for a specific device. Finally, the user can also set the skip parameter which defines the number of database entries to be ignored since the beginning and the limit parameter which defines the maximum number of entries returned. All these filters can be used separately or together. The response is a list of messages according to the format presented in code 2.

```
message MeasurementResponse{
    string DeviceId = 1;
    string Field = 2;
    string Value = 3;
    string Date = 4;
}
```

Code 2: Proto message MeasurementResponse.

4.4 Market Microservice

The market microservice, unlike all other microservices, is divided into two parts: the first part (Energy Trading Algorithms) is responsible for allocating electricity to the community members using a market algorithm, while the second part (Market API) allows interaction with external entities, both for obtaining and updating the energy matches and also for modifying the price profiles of each community member.

4.4.1 Energy Trading Algorithms

The microservice in this part is responsible for processing the measurements that the smart meters send to the meters microservice every 15 minutes, so that this data is processed every 15 minutes with a 5 minutes tolerance window for sending the readings. This means for example that the 12:00 hours readings will only be processed at 12:05. This five minutes interval allows most smart meters to send their readings, especially those in places where the mobile network connectivity is unstable, such as the smart meter installed in "Casa da Cultura de Ílhavo" building, which is located underground.

The service works via a *cron job* that runs the program at each time interval. It starts by identifying the last timestamp it handled and then processes the data from that moment until the current timestamp. At each interval, requests are made to the meter microservice using the *RetrieveMeasurement()* method with a filter consisting of the last processed timestamp and a limit of two. This returns two readings for each smart meter present. From the returned readings, the "Active import" and "Active export" readings are grouped into two lists which are used to check whether there are two consecutive readings for each smart meter and whether their timestamps correspond to the interval currently being processed. All smart meters that do not meet these two conditions are excluded from the process for that time interval.

Once the two lists have been organized and filtered, a bid-like object is created from them, according to code 3 format, corresponding to the difference between the amount of energy exported and the amount of energy imported in this interval. Both the energy exported and the energy imported in an interval are calculated from the difference between the last reading in the list (corresponding to the timestamp to be processed) and the reading from the previous period (timestamp to be processed minus 15 minutes). If the energy parameter in the bid is greater than zero, this bid will be added to a list of bids corresponding to a list of energy sales orders, while if it is below zero, the bid will be added to a list of asks corresponding to a list of purchase orders. At the same time as the lists are created, the buying and selling price defined for the time period to be processed is consulted in the community member's profile.

```
class Bid:
    id: int
    sell_price: float
    buy_price: float
    consumption: float
    production: float
    energy: float
    timestamp: datetime.datetime
```

Code 3: Bid object format.

The lists of asks and bids are processed by an algorithm to create energy matches. In this part, nine algorithms have been implemented with the aim of exploring different possibilities for exchanges within energy communities. Some of these algorithms require an auxiliary function that determines the point of intersection between the lists of bids and asks; the equilibrium point has different purposes for different algorithms. The intersection of the two lists is contained in one of the nine situations presented in figure 4.5, or in a special situation where the two lists are empty. Situations 1, 2, 3 and 4 are cases in which the two lists have a real intersection at some point. Situation 5, on the other hand, represents a case in which there is no possible intersection, since all the asks are lower than the bids, meaning that in some market algorithms there has been no exchange of energy within the community and that the prices of the REC members are not aligned to each others' expectations. In situations 6 and 7, there is the possibility of finding a virtual intersection point, which corresponds to the midpoint between the last point on the shortest list and the corresponding point on the other list. Finally, in situations 8 and 9, there is no intersection point because one of the lists is empty.

After processing these lists, all market algorithms return a final series of computed energy exchanges/matches for the specific period, either between members of the community or with the public grid. All the matches that involve buying or selling energy to the public grid have the structure presented in code 4. For the case of the community energy exchanges, in addition to the parameters in code 4 there is also a metadata section, which initially consists of the *match_id*, *transaction_state* and *updated_at*. The *transaction_state* represents the state of the transaction payment process, and it can hold the following values: *created*; *sent*; *paid*; *not paid*; or *error*. The *created* state corresponds to the initial state of the transaction creation. The *sent* state indicates that the transaction has been sent to the smart meter for processing. The *paid* state means that the transaction has been paid without any problems. On the other hand, the *not paid* state means that the transaction payment process was not completed successfully. The *error* state means that an error occurred during the whole process of creating the transaction object. These states are only changed by the second part of this microservice, as well as any updates made to the matches.

All the implemented algorithms have been created in such a way that there is no need to change anything in the other parts of the project, either internally or externally. All that is necessary is to change the name of the algorithm to be called. The next nine subsections describe the implemented algorithms.

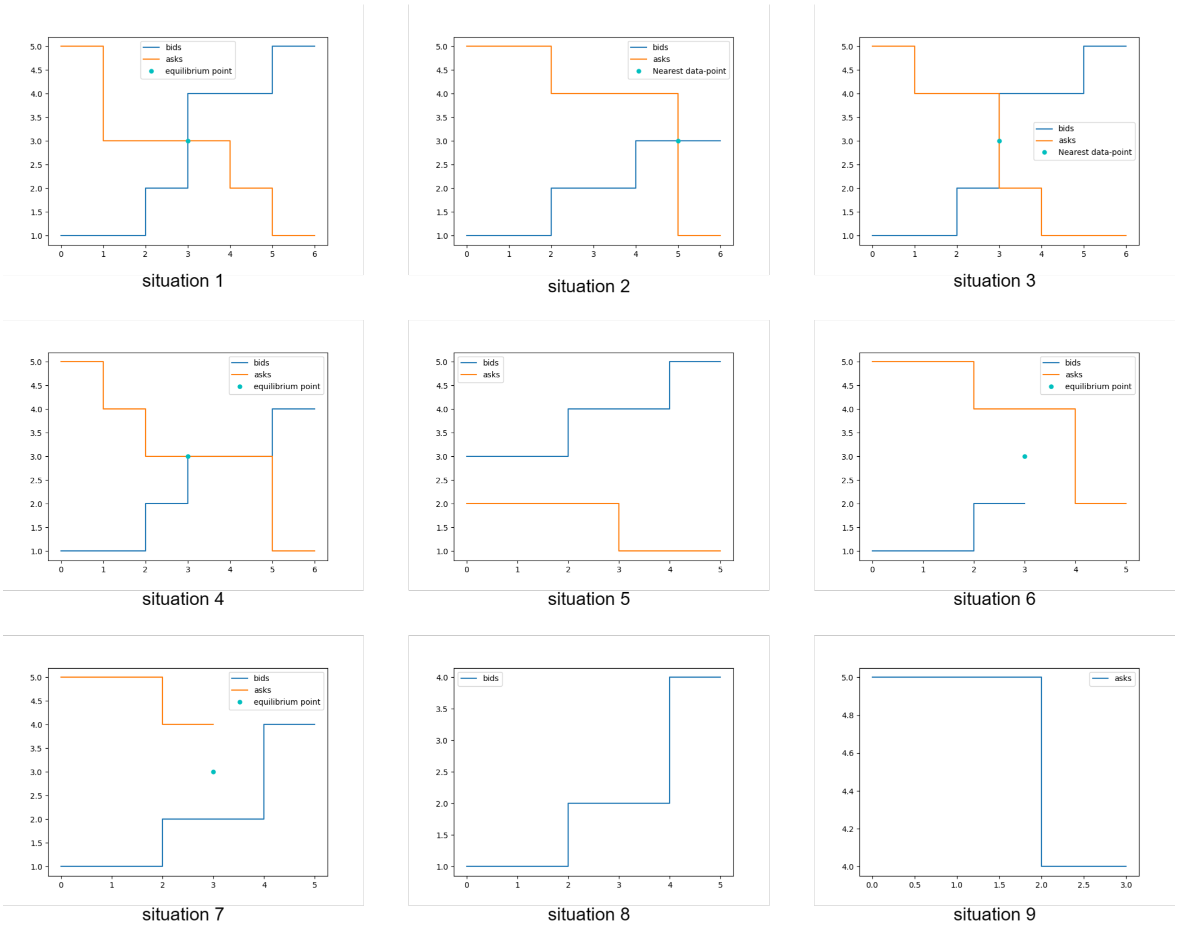

Figure 4.5: Possible scenarios for equilibrium point calculation in market algorithms.

```
match = {
    "timestamp": timestamp,
    "buyer_id": buyer_id,
    "seller_id": seller_id,
    "energy": energy,
    "price": price,
    "created_at": time,
    "_id": match_id,
}
```

Code 4: Match object format.

Public Grid

This method represents a system where there is no community, where all the energy transacted by each potential REC member is bought and sold to the public grid. A fixed purchase price of 0.1624€/kWh and sale price of 0.03€/kWh were considered for this algorithm based on the regulated energy market at a simple tariff for normal low voltage customers (<=20.7 kVA and >2.3 kVA). The purchase price is taken from Directive 3/2023 [29] of January 11. The selling price, on the other hand, is an approximation obtained from the various offers made by electricity market operators.

Static Price

In this method, all members of the community who wish to purchase energy during that period are served with energy produced within the community rather than energy from the public grid. Buyers are served in a predetermined order, such as the order of membership in the community, until there are no more buyers to serve or until there is no more energy coming from the community. In cases where there is not enough energy within the community to satisfy all consumers, the deficit is bought from the public grid. In the opposite case, the surplus energy is sold to the public grid. Based on the prices for the public grid method, the price for buying and selling energy to the public grid within the community was established to be 0.12€/kWh.

Uniform Price

In this case, all the requests to buy and sell energy are organized and the intersection point is found, as described above. The intersection point acts as an equilibrium point in this case, as it defines the price of the energy to be traded within the community and separates the transactions that take place within the community from those that take place with the public network. All orders to the left of the equilibrium point will trade energy within the community. The rest will be allocated to the public network. The exchange of energy within the community will take place in the order of the two lists of requests, i.e. the first buyer will buy energy from the first seller, and if it doesn't have enough energy to satisfy his request, it will buy the rest from the next seller on the list (figure 4.6). In order to make some preliminary tests and validate the current implementation of this and the following algorithms, some price values for buying and selling energy within the community were considered for the REC members according to a random distribution in a price interval with an average value of 0.12€/kWh (the same as for the *Static Price* scenario).

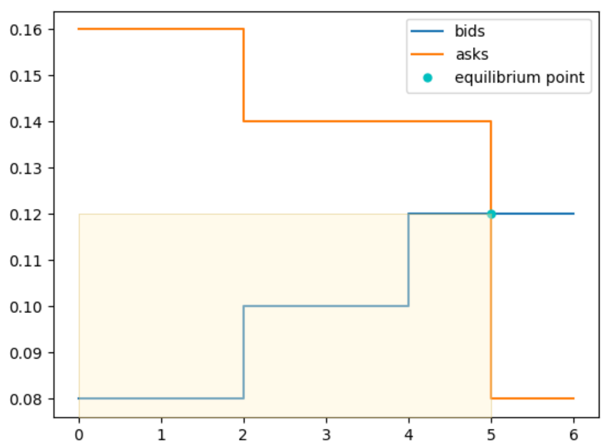

Figure 4.6: Uniform price market algorithm implementation.

Buyers Price

This case works in the same way as the previous one (*Uniform Price*), except that the price within the community depends on each order up to the point of intersection, which means that the price defined in each order is the price of the quantity of energy in that order (figure 4.7). For example, if there is an order for 1 kWh at X cents and a sale of 1 kWh at Y cents (X>Y), the buyer will pay the price it proposed, i.e. X cents.

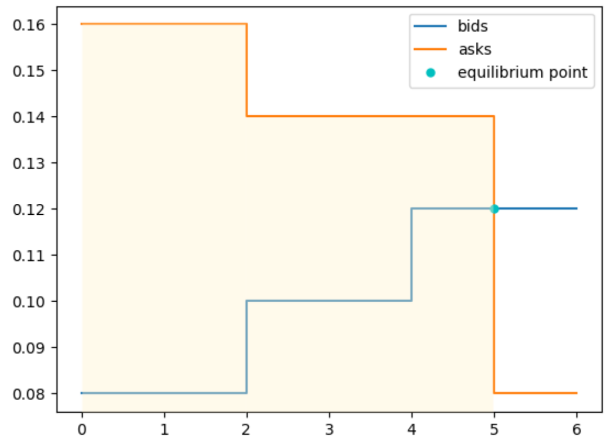

Figure 4.7: Buyers price market algorithm implementation.

Sellers Price

This method works in the same way as the *Buyers Price* method, with the small difference that the price is set by sales orders rather than purchase orders (figure 4.8). For example, if there is an order for 1kW at X cents and a sale of 1kW at Y cents (X>Y), the buyer will pay the price suggested by the seller, i.e. Y cents.

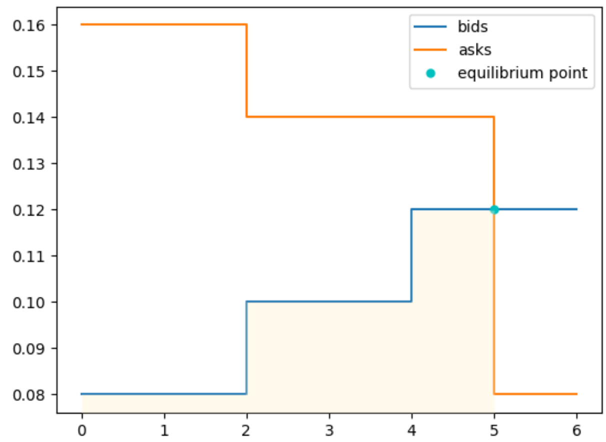

Figure 4.8: Sellers price market algorithm implementation.

Average Price

This method uses the same assumptions as the previous three, except that the energy price is calculated using the average between the selling and buying prices of their respective orders (figure 4.9). The price of the energy exchanged within each pair of matched sellers and buyers in the list is calculated using the average of the prices set by both. Assuming the above two cases, the price charged will be (X+Y)/2.

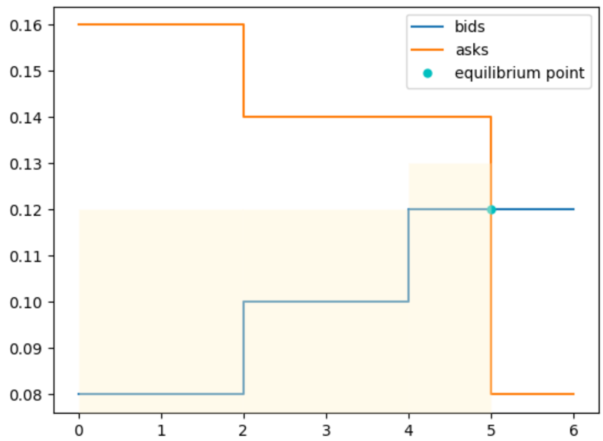

Figure 4.9: Average price market algorithm implementation.

Single-Sided Auction

In this method, sales orders are organized in the same way as before, that is, in ascending order of sales price. On the other hand, purchase orders are not ranked, but only the total energy demand of these purchase orders is calculated. The energy price is calculated taking into account the total energy demand of the orders and the total energy to be sold, in one of the following ways:

- When the total demand for energy in the purchase orders is less than the total energy to be sold within the community, the energy price corresponds to the price of the sales order where the demand for energy in the purchase orders is met, i.e. the order in the list of sales orders where the sum of the energy of all the orders up to and including it is equal to or greater than the demand present in the purchase orders.
- When the total demand is higher, the energy price is equal to the price of the last sales order on the ordered list. With the price found, each purchase order will buy a proportional amount of energy within the community and the rest from the public grid. For example, if the community has 80% of the energy needed to supply the purchase orders for sale, each purchase order will buy 80% of the energy it wants from the community and 20% from the public grid.

All the energy that is lacking or in excess in the community is bought from or sold to the public grid based on the prices of the *Public Grid* method.

Bill Sharing

In this method, the community shares the energy it produces and each member buys the rest from the public grid. In this case, the energy is distributed free of charge to all members of the community. If there is surplus energy within the community, it is sold to the public grid, i.e. all offers with surplus energy after the exchange within the community will sell the unused surplus to the public grid. On the other hand, if there is a shortage of energy within the community, all requests with a shortage of energy will buy the percentage of the shortage within the community from the public grid. In the exceptional case where the amount of energy produced and consumed within the community is equal, the amount of money transacted in that period is zero.

Mid-Market Rate

In this case, the price of energy within the community is defined by the average between the purchase price and the sale price to the public grid, (0.1624 + 0.03) / 2 = 0.0962 €/kWh, taking into account the same prices used in the *Public Grid* method. As in the previous case (*Bill Sharing*), the existence of a deficit or surplus of energy in the community is verified and treated in the same way. The energy that was shared free of charge within the community in the previous case is now exchanged between all members at the price defined above.

4.4.2 Market API

The second part of this microservice consists on the interface through which external entities or other microservices can interact with it from the outside and consume and modify matches' data and price profiles. This is split into two modules, the first called *AuthenticatedMarketService*, which is responsible for the interaction between this microservice and the Transactions microservice. For that purpose, the following methods can be used:

- *RetrieveMatches()*, through which a query on the matches stored in the market microservice can be performed. This can be done in conjunction with a filter that allows the selection of various optional parameters. The query can for instance filter by buyer or seller ID, by transaction status or multiple statuses. It can also filter just one match by its ID, select multiple matches present in a time interval, or even define the skip and limit parameters of the query. In the end, the filter should have a structure like the one in code 5.

```
message MatchesFilter{
    optional string buyerID = 1;
    optional string sellerID = 2;
    repeated State state = 3;
    optional string startTimestamp = 4;
    optional string endTimestamp = 5;
    optional string matchID = 6;
    optional int32 skip = 7;
    optional int32 limit = 8;
}
```

Code 5: Match filter format.

- *UpdateMatch()* performs the most critical operations of the whole microservice, since it through here that the status of a match is updated, the *transactionID* is added to it, or a message is included to explain the occurrence of an event in the transaction process. This method is the real reason for creating an independent module with internal authentication to the market service, since only the transactions microservice, which is responsible for all interactions with transactions, should be able to access it. When interacting with this module, it is possible to send a list of several matches to be modified, composed by several messages according to the format presented in code 6.

```
message UpdateMatch{
    string matchID = 1;
    State state = 3;
    optional string transactionID = 2;
    optional string message = 4;
}
```

Code 6: Match update format.

The second module called *MarketService* is used for communicating with external entities, for instance enabling community members to check their matches, the status of their transactions and also visualize and define their price profiles. This is divided into three methods:

- *GetMatches()* works in the same way as the *RetrieveMatches()* method, but without the ability to modify the matches. The filters to be applied are the same, and the end result is a list of matches with the format of code 7.

```
message MatchResponse{
    string timestamp = 1;
    string buyerID = 2;
    string sellerID = 3;
    float energy = 4;
    float price = 5;
    string id = 6;
    string createdAt = 7;
    optional string transactionID = 8;
    optional State transactionState = 9;
    optional string updatedAt = 10;
    optional string message = 11;
}
```

Code 7: Match response format.

- *UpdatePrice()* allows users to add or change the price profiles of each REC member. This method works by inserting a prices list with the format of code 8, modifying the timestamps that already have prices defined and adding the prices for timestamps that don't have prices defined.
- *GetPrices()* returns a list of messages according to the format displayed in code 8. For that purpose, a filter like the one presented in code 9 must be used, which is composed of a meter ID and some optional parameters, including the possibility of defining the interval to be filtered and the skip and limit parameters of the query.

```
message PriceUpdate{
    string timestamp = 1;
    string meterID = 2;
    float buyPrice = 3;
    float sellPrice = 4;
}
```

Code 8: Price update format.

```
message PriceFilter{
    string meterID = 1;
    optional string startTimestamp = 2;
    optional string endTimestamp = 3;
    optional int32 skip = 4;
    optional int32 limit = 5;
}
```

Code 9: Price filter format.

CHAPTER 5

Results

This chapter presents the results obtained using an energy dataset, with the aim of comparing the implemented algorithms. It also shows how the implemented system works with data obtained in real-time from smart meters deployed in the field.

5.1 Market results for electricity dataset

In order to analyze the operation of the implemented algorithms, it was necessary to use an external dataset, as the number of smart meters installed in the field is small and would not be representative of a REC in operation. With this in mind, an electricity dataset [30] available in the literature was selected, which has already been used in other parts of the COMSOLVE project and is described in detail in [31]. The dataset consists of data from 51 buildings, 50 residential buildings and one public building, of which 15 buildings are both consumers and producers of energy (prosumers). Although the dataset consists of several data tables, only the "Total Consumers" and "Total Producers" tables are used, which contain the consumption and production data for each of the 51 buildings.The whole dataset consists of one year of electricity measurements, for which the results below reflect the total numbers considering this entire period.

Price profiles were created for each of the 51 buildings, one for buying and one for selling. These profiles were randomly generated using Python's random function to produce a value between 0.10€/kWh and 0.14€/kWh. For each building, one value was generated for the purchase profile and another for the production profile. The consumption and production profiles are therefore independent and as a result, the consumption price can be the same as of the production price, higher or lower. There is no link between the generation of the two values, only that they are randomly generated in the same way. Each building is associated with an energy purchase price and an energy sale price, which are fixed over the total simulation time and used throughout all the tests performed using the dataset. This means that for the system evaluation, it was assumed that all REC members have a simple tariff (no multiple nor dynamic tariffs were considered), but with different purchase and sales prices based on the values obtained through the randomization process.

With this in mind, the electricity data from the dataset was passed through all the implemented algorithms and stored in independent collections (a MongoDB structure that corresponds to a table in a relational database). With the consumption, production and price data, bid objects were generated with the structure presented in code 3 and processed as described in the market microservice implementation for the REC formed by the 51 buildings of the dataset.

Figure 5.1 shows the electricity source for each of the algorithms, by analyzing the percentage of energy exchanged inside and outside the community. The blue color represents the energy exchanged with the public grid and green represents the energy exchanged within the community. The darker blue and green colors represent the energy bought from the public grid and from the community, respectively, while the lighter colors represent the energy sold to them. This shows that the *static*, *single_sided*, *bill_sharing* and *mid_market_rate* algorithms exchange the same amount of energy among community members. The same happens with the *uniform*, *buyers_price*, *sellers_price* and *average_price* algorithms. This doesn't mean that they all exchange energy in the same way, but rather that the total amount of energy exchanged inside the REC over the course of a year is the same. The second group of algorithms is the only one where the energy exchange is the same regardless of which algorithm is used, because the matching process is the same for all four, only the pricing process is different. On the other hand, the first group of algorithms all work differently because they all have different ways of matching and pricing.

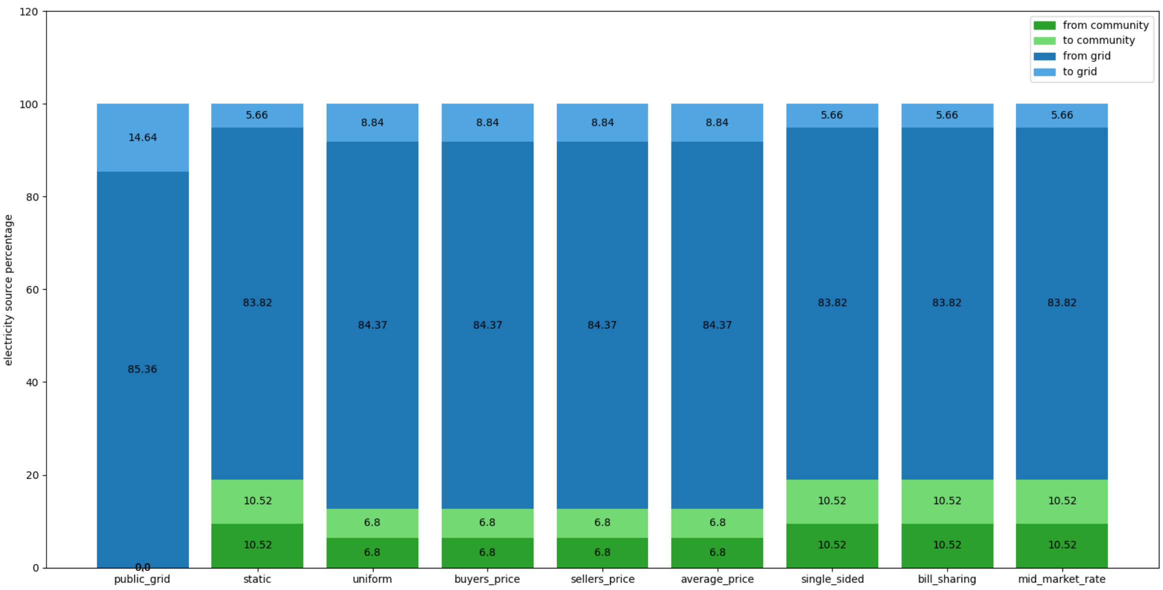

Figure 5.1: Electricity source for different market algorithms.

Figure 5.2 shows the evolution of the electricity source over the different months of the year.

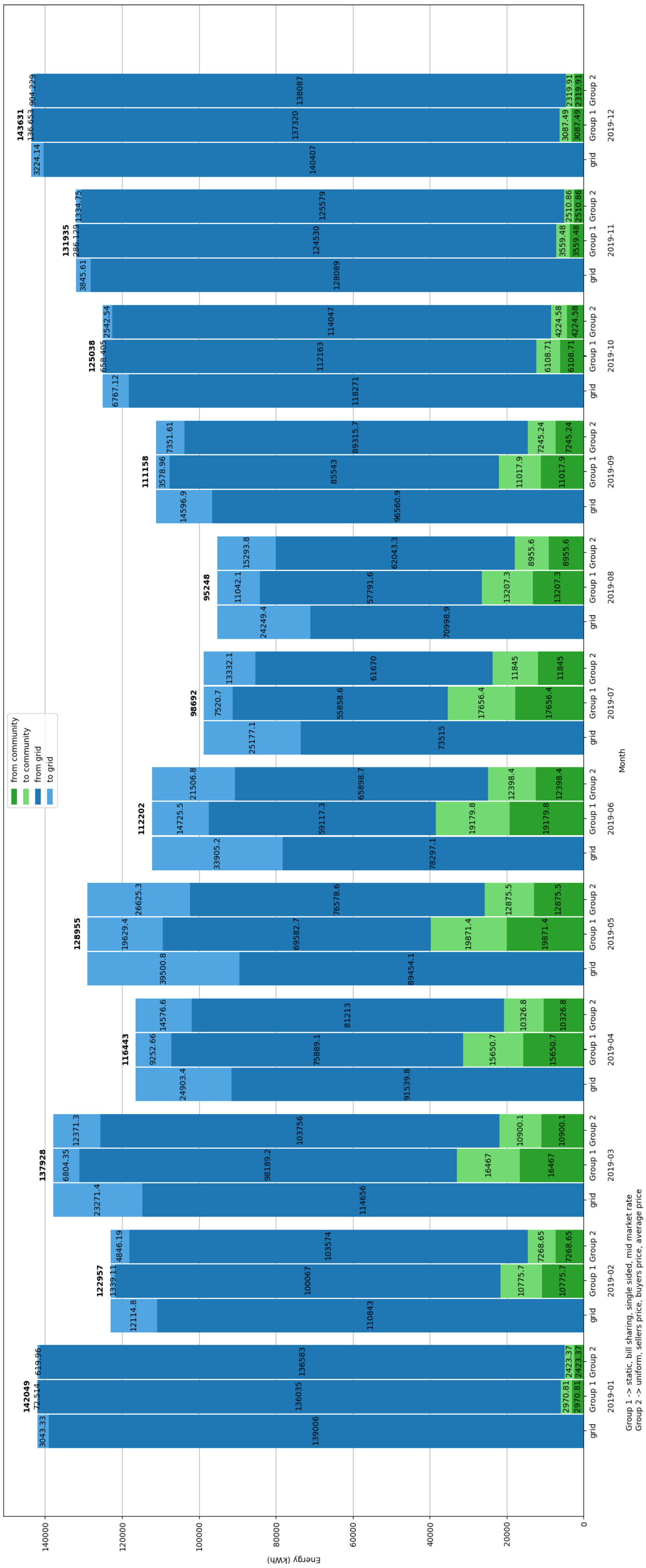

Figure 5.2: Electricity source by month for the *public_grid* and the two groups of algorithms.

Based on the two groups of algorithms defined above, it is possible to visualize the tendency for the energy exchange within the community to increase during the middle of the year, which corresponds to the Summer period, the time of the year when the photovoltaic panels produce more energy. However, it can also be seen that energy consumption is higher in the opposite months, when temperatures are lower and more electricity is needed on a daily basis.

To explain the differences between the results of both groups, one can analyze the total number of matches per algorithm as presented in figure 5.3. One can see that the four algorithms in the first group (*uniform*, *buyers_price*, *sellers_price* and *average_market*) have exactly the same number of price matches among REC members in each situation, and the second group (*static*, *single_sided*, *bill_sharing* and *mid_market_rate*) does not.

In addition, figure 5.3 compares the total number of matches for each algorithm in both groups with the total number of matches in the scenario with no REC, using the public grid only (*public_grid* algorithm), with the percentage difference in the number of matches relatively to the public grid scenario shown at the top of each bar. Each bar also shows the percentage of matches exchanged with the public grid in blue, and the percentage of matches within the community in green. Besides observing that the four algorithms in the first group work in the same way, it is possible to see that the *bill_sharing* algorithm does not produce any energy matches within the community, since this energy is transferred within the community at zero cost. It can also be seen that only the *single_sided* and *mid_market_rate* algorithms produce more matches than the baseline scenario (*public_grid* algorithm). This is due to the fact that, in those cases, the energy produced within the community is distributed in a proportional way to everyone in the REC, unlike other algorithms, such as those in the second group, which carry out these exchanges taking into account the equilibrium point.

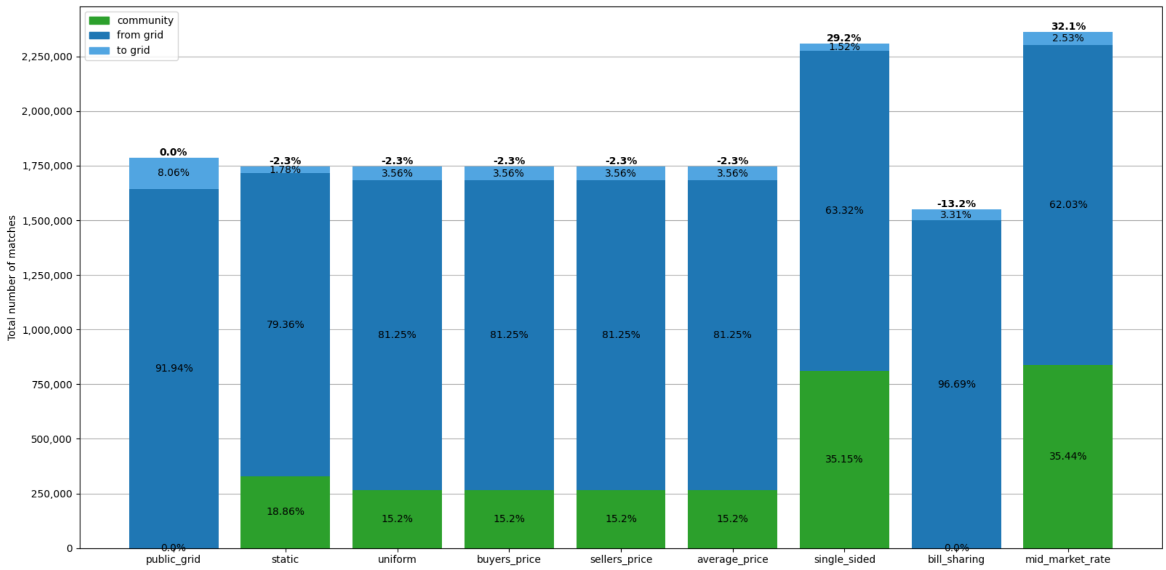

Figure 5.3: Total number of matches for different market algorithms.

Figure 5.4 shows the total monetary values exchanged by the community over the course of a year for each algorithm. Blue represents the values exchanged with the public grid, dark green corresponds to the values within the community and light green represents the net amount, i.e. the difference between the buy and sell values. The sell bar represents the total value of energy sold by the community, both inside and outside the community (public grid). The buy bar represents the opposite, that is the total value bought by the community, both inside and outside the community. Finally, the bar on the right represents, in light green, the difference between the sell and buy bars. Regarding the net value, it is worth to mention that it results from the balance between the sell and buy operations with the public grid, since the values exchanged within the community cancel each other out.

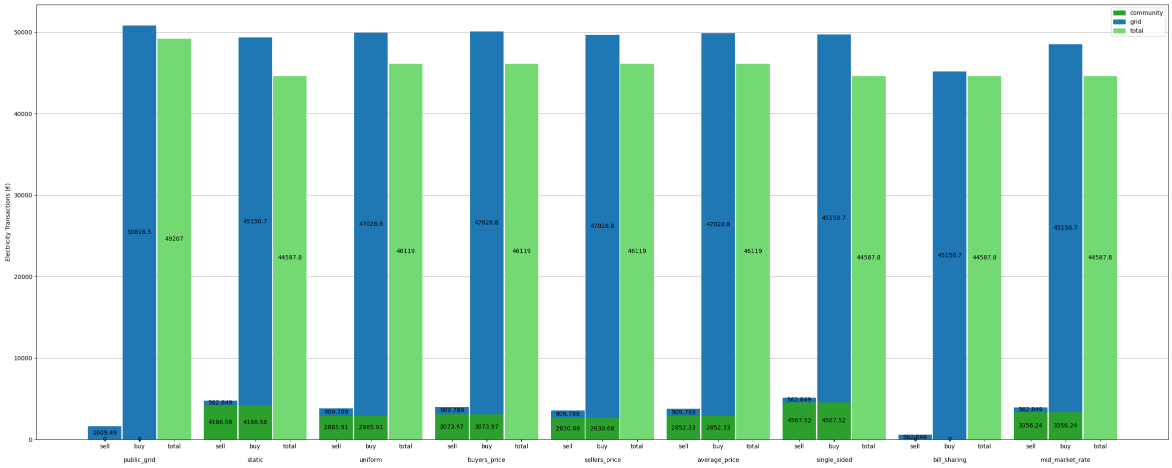

Figure 5.4: Total monetary values for different market algorithms.

In order to better compare the different algorithms, another parameter was taken into account, namely the average price value of buying and selling energy for the REC members, which can be seen in figure 5.5. The blue bars represent the average price of selling energy and the green bar represents the average price of buying energy. Both prices include transactions within the community and with the public grid. Looking at the graph, one algorithm stands out, *bill_sharing*, because it is the one with the lowest average energy purchase price (0.1443 €/kWh), but it also stands out because its average sales price is the lowest (0.0105 €/kWh). Another highlight is associated with the *single_sided* algorithm, which is the one that most values the surplus within the community, with the highest average energy selling price of all algorithms (0.0956 €/kWh).

An important detail in all algorithms is that all pricing takes into account the public grid prices, and the prices within the community must always be within the public grid price range. This means that the buy price is lower than the buy price on the public grid and the sell price is higher than the sell price on the public grid. With the exception of the "bill_sharing" algorithm, where exchanges within the community are made at zero cost, as mentioned above.

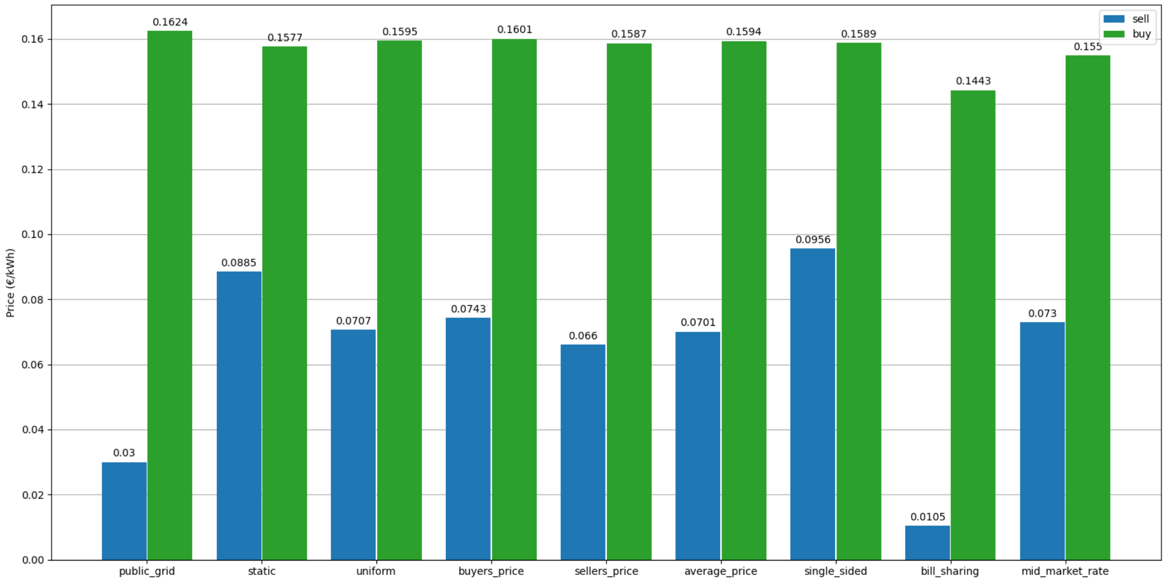

Figure 5.5: Average electricity prices for different market algorithms.

5.2 Market operation using real-time data

In order to demonstrate how the implemented system works in real-time, two smart meters, installed in the field under the scope of COMSOLVE project, were considered. The first one was deployed in the office of one of the project partners (Digitalmente) with the ID *es-sms-15*, where there is only electricity consumption, while the second one was installed in a household in Ílhavo with electricity generation through photovoltaic panels (ID *es-sms-18*). Figure 5.6 shows the energy consumption and production values for both buildings in two different timestamps: "2023-10-09T14:00:00Z" and "2023-10-09T14:15:00Z". Looking at the consumption figures for each, one can see that building *es-sms-15* has consumed 352 Wh (479567 - 4798215), while building *es-sms-18* has consumed 15 Wh (21435216 - 21435201) in the interval shown. When it comes to production, the *es-sms-15* building has a value of zero because it does not include any energy production source, while the building *es-sms-18* has a production of 366 Wh (3936678 - 3996312). This means that the *es-sms-15* building had an energy deficit of 352 Wh, while the *es-sms-18* building had an energy surplus of 351 Wh (366 - 15).

By using the *mid_market_rate* algorithm in the market microservice and processing the measurements shown in figure 5.6, the matches shown in figure 5.7 are generated. These represent the purchase by building *es-sms-15* of the 351 Wh surplus from building *es-sms-18*. In this particular timestamp, since building *es-sms-15* is not able to fulfill all its energy needs from the surplus of building *es-sms-18*, and since there are no other REC members in this test setup, it must buy the remaining electricity from the public grid, which is represented by the match with *sellerID* 100 (ID assigned in the system to the public grid operations). In the displayed matches, one can observe the price per kWh of each match: the match with the public grid has a price of 0.1624 €/kWh and the match with *es-sms-18* building is 0.0962 €/kWh. The timestamp of the creation of the matches is also shown, with the value of

table mean	_measurement group string	_field group string	_value no group double	_start group dateTime:RFC3339	_stop group dateTime:RFC3339	_time no group dateTime:RFC3339	device group string
0	energy	Active export	0	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:00:05.000Z	es-sms-15
0	energy	Active export	0	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:15:05.000Z	es-sms-15
1	energy	Active import	4798215	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:00:05.000Z	es-sms-15
1	energy	Active import	4798567	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:15:05.000Z	es-sms-15
2	energy	Active export	3936312	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:00:05.000Z	es-sms-18
2	energy	Active export	3936678	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:15:05.000Z	es-sms-18
3	energy	Active import	21435201	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:00:05.000Z	es-sms-18
3	energy	Active import	21435216	2023-10-09T14:00:00.000Z	2023-10-09T14:30:00.000Z	2023-10-09T14:15:05.000Z	es-sms-18

Figure 5.6: Example of electricity measurements stored in the meters microservice database.

"2023-10-09T14:20:02Z" for both cases. In the case of the match between the two buildings belonging to the community, whose transaction has already been sent for payment by the smart meter, the additional field *transactionState* is present with status "Sent".


```
"matches": [
    {
        "timestamp": "2023-10-09T14:15:00Z",
        "buyerID": "es-sms-15",
        "sellerID": "100",
        "energy": 0.0010000000474974513,
        "price": 0.1624000072479248,
        "id": "65240c12f21d75d564dce094",
        "createdAt": "2023-10-09T14:20:02Z"
    },
    {
        "timestamp": "2023-10-09T14:15:00Z",
        "buyerID": "es-sms-15",
        "sellerID": "es-sms-18",
        "energy": 0.35100001096725464,
        "price": 0.09619999676942825,
        "id": "65240c12f21d75d564dce095",
        "createdAt": "2023-10-09T14:20:02Z",
        "transactionState": "Sent"
    }
]
```


Figure 5.7: Example of the matches for the timestamp "2023-10-09T14:15:00Z".

CHAPTER 6

Conclusion

This chapter summarizes all the work carried out, as well as a final analysis of the results obtained for the local market system. It also presents some points for possible future work within the scope of this project.

6.1 Conclusions

The main objective of this thesis was to develop a market system for the electricity transactions in a REC. The basic concepts of energy trading systems and RECs are presented. It describes the transformation processes that the energy system is undergoing as it becomes more decentralized and consumer focused. An explanation of the types of strategies that exist for an energy trading system is then provided, presenting possible bidding strategies and market clearing approaches. It is followed by a description of the RECs concept, detailing their objectives, why they were created and for what purpose. Ways of organizing RECs are also presented, as well as some trading algorithms for conducting energy exchanges.

Next, the COMSOLVE project, in which this thesis is integrated, is explained, describing its general architecture and some of the systems within it. Following that, an architecture for the market system is outlined, based on a set of requirements and use cases associated with both the main project and this thesis alone.

In the course of the project and the development of a real-time electricity monitoring system using a custom 5G smart meter, an implementation was made first for low-voltage systems and later for medium-voltage and special low-voltage systems, since this real-time monitoring system was required to be deployed in a group of buildings supplied with very distinct voltage distribution levels.

At the same time, the architecture of the system was developed and the various microservices were implemented. When it came to the market microservice, it was structured so that different types of energy matching algorithms could be tested, and these were implemented at the same time as of the microservice.

Finally, several tests were performed to validate the operation of the deployed system, as well as of the implemented matching algorithms in order to compare the resulting market metrics.

When analyzing the results obtained by the different market algorithms, there are several ones that stand out in different ways. Starting with the *uniform*, *buyers_price*, *sellers_price* and *average_price* algorithms, it can be seen that they all work in the same way when it comes to distributing energy among the members of a community, and that the price profiles they define are very important in this distribution, since the final price of the energy exchanged in each situation is what differentiates them. The *single_sided* algorithm gives equal importance to the price profiles, but only to those defined by the energy sellers, which allows the average energy sales prices within the community to be higher than in the other algorithms. On the other hand, the *static* and *mid_market_rate* algorithms do not take into account the price profiles defined by the members and apply the same price for the energy exchanged, with the *mid_market_rate* algorithm using the public grid prices as a reference and the *static* algorithm using the price defined by the REC management entity. Finally, the *bill_sharing* algorithm is characterized by the fact that there is no exchange of energy within the community, since the entire REC shares the energy it produces free of charge.

Looking at all the algorithms presented, it's not possible to say that one is better than the other, but rather that depending on the type of community model the members prefer, one is better suited to the situation than the other. For example, the *bill_sharing* algorithm will work better in a condominium where there is no owner of the panels but they are shared by all members, or even in the case of a social housing estate where there is an installation of panels to help reduce the energy bills of its members. As for the other algorithms, their use should also be considered by taking into account the type of community the members want, whether it's a community where its members define their own energy prices through price profiles, or a community where the aim is simply to reduce the energy bill at the end of the month without worrying about creating price profiles.

In summary, each of the algorithms presented is beneficial, as they all help to reduce the energy bill, with the worst case being that members who, because of the prices they set, have to pay the same amount as they would if they were not participating in a REC. The whole system is designed to reduce energy costs and encourage people to use systems like this, reducing their energy dependency and making the energy market as consumer focused as possible.

It is therefore possible to conclude that all the objectives of this thesis have been achieved, but it is possible to continue and improve the methods used, as well as to compare them with other similar systems.

6.2 Future Work

With the results obtained, it is possible to conclude that further work can be carried out, including:

- Test the system with different tariffs for the public grid prices, such as bi-hourly tariffs or even dynamic pricing;
- Implement more algorithms and create some variations of those already implemented;
- Execute more performance evaluation with real buildings, a community with different types of members is ideal;
- Use more metrics to compare algorithms, such as the ones employed by Khorasany et al. [32];
- Implement a method that allows different algorithms to be used within the same community at the same time, e.g. by having members organized in different clusters, in order to tackle problems such as energy poverty.

References

[1] P. Pinson, T. Baroche, F. Moret, T. Sousa, E. Sorin, and S. You, "The emergence of consumer-centric electricity markets," *Distribution & Utilization*, vol. 34, no. 12, pp. 27–31, 2017.

[2] T. D. Couture, K. Cory, C. Kreycik, and E. Williams, "Policymaker's guide to feed-in tariff policy design," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2010.

[3] G. C. Christoforidis, I. P. Panapakidis, T. A. Papadopoulos, *et al.*, "A model for the assessment of different net-metering policies," *Energies*, vol. 9, no. 4, p. 262, 2016.

[4] H. Muhsen, A. Allahham, A. Al-Halhouli, M. Al-Mahmodi, A. Alkhraibat, and M. Hamdan, "Business Model of Peer-to-Peer Energy Trading: A Review of Literature," *Sustainability*, vol. 14, no. 3, p. 1616, 2022.

[5] Y. Jin, J. Choi, and D. Won, "Pricing and Operation Strategy for Peer-to-Peer Energy Trading Using Distribution System Usage Charge and Game Theoretic Model," *IEEE Access*, vol. 8, pp. 137 720–137 730, 2020.

[6] K. Chen, J. Lin, and Y. Song, "Trading strategy optimization for a prosumer in continuous double auction-based peer-to-peer market: A prediction-integration model," *Applied energy*, vol. 242, pp. 1121–1133, 2019.

[7] J. Wang, Q. Wang, N. Zhou, and Y. Chi, "A novel electricity transaction mode of microgrids based on blockchain and continuous double auction," *Energies*, vol. 10, no. 12, p. 1971, 2017.

[8] P. Vytelingum, S. D. Ramchurn, T. D. Voice, A. Rogers, and N. R. Jennings, "Trading agents for the smart electricity grid," in *The Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010) (10/05/10 - 14/05/10)*, Event Dates: May 10-14, 2010, 2010, pp. 897–904. [Online]. Available: `https://eprints.soton.ac.uk/268361/`.

[9] H. Huang, S. Nie, J. Lin, Y. Wang, and J. Dong, "Optimization of peer-to-peer power trading in a microgrid with distributed PV and battery energy storage systems," *Sustainability*, vol. 12, no. 3, p. 923, 2020.

[10] C. H. Leong, C. Gu, and F. Li, "Auction mechanism for P2P local energy trading considering physical constraints," *Energy Procedia*, vol. 158, pp. 6613–6618, 2019.

[11] J. Lin, M. Pipattanasomporn, and S. Rahman, "Comparative analysis of auction mechanisms and bidding strategies for P2P solar transactive energy markets," *Applied energy*, vol. 255, p. 113 687, 2019.

[12] G. C. Okwuibe, M. Wadhwa, T. Brenner, P. Tzscheutschler, and T. Hamacher, "Intelligent bidding strategies in local electricity markets: A simulation-based analysis," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, IEEE, 2020, pp. 1–7.

[13] H. Zang and J. Kim, "Reinforcement learning based peer-to-peer energy trade management using community energy storage in local energy market," *Energies*, vol. 14, no. 14, p. 4131, 2021.

[14] M. Khorasany, Y. Mishra, and G. Ledwich, "Market framework for local energy trading: A review of potential designs and market clearing approaches," *IET Generation, Transmission & Distribution*, vol. 12, no. 22, pp. 5899–5908, 2018.

[15] Z. Zhou, W. K. Chan, and J. H. Chow, "Agent-based simulation of electricity markets: a survey of tools," *Artificial Intelligence Review*, vol. 28, pp. 305–342, 2007.

[16] W. Tushar, C. Yuen, H. Mohsenian-Rad, T. Saha, H. V. Poor, and K. L. Wood, "Transforming energy networks via peer-to-peer energy trading: The potential of game-theoretic approaches," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 90–111, 2018.

[17] W. Tushar, T. K. Saha, C. Yuen, *et al.*, "A motivational game-theoretic approach for peer-to-peer energy trading in the smart grid," *Applied energy*, vol. 243, pp. 10–20, 2019.

[18] C. Van Dinther, *Adaptive bidding in single-sided auctions under uncertainty: an agent-based approach in market engineering*. Springer Science & Business Media, 2007.

[19] A. Elmouatamid, R. Ouladsine, M. Bakhouya, N. El Kamoun, M. Khaidar, and K. Zine-Dine, "Review of control and energy management approaches in micro-grid systems," *Energies*, vol. 14, no. 1, p. 168, 2020.

[20] D. Y. Yamashita, I. Vechiu, and J.-P. Gaubert, "A review of hierarchical control for building microgrids," *Renewable and Sustainable Energy Reviews*, vol. 118, p. 109 523, 2020.

[21] L. Meng, E. R. Sanseverino, A. Luna, T. Dragicevic, J. C. Vasquez, and J. M. Guerrero, "Microgrid supervisory controllers and energy management systems: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 1263–1273, 2016.

[22] P. Tian, X. Xiao, K. Wang, and R. Ding, "A hierarchical energy management system based on hierarchical optimization for microgrid community economic operation," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2230–2241, 2015.

[23] T. Sousa, T. Soares, P. Pinson, F. Moret, T. Baroche, and E. Sorin, "Peer-to-peer and community-based markets: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 104, pp. 367–378, 2019.

[24] G. Santos, P. Faria, Z. Vale, T. Pinto, and J. M. Corchado, "Constrained generation bids in local electricity markets: a semantic approach," *Energies*, vol. 13, no. 15, p. 3990, 2020.

[25] C. Long, J. Wu, C. Zhang, L. Thomas, M. Cheng, and N. Jenkins, "Peer-to-peer energy trading in a community microgrid," in *2017 IEEE power & energy society general meeting*, IEEE, 2017, pp. 1–5.

[26] [Online]. Available: `https://comsolve.pt/`.

[27] [Online]. Available: `https://www.e-redes.pt/pt-pt/como-posso-aceder-porta-han`.

[28] [Online]. Available: `https://www.e-redes.pt/sites/eredes/files/2020-07/DEF-C44-509.pdf`.

[29] [Online]. Available: `https://www.erse.pt/media/iahipekm/diretiva-3_2023.pdf`.

[30] C. Goncalves, R. Barreto, P. Faria, L. Gomes, and Z. Vale, *Dataset of an energy community's consumption and generation with appliance allocation for one year*, 2022. [Online]. Available: `https://zenodo.org/records/6778401`.

[31] ——, "Dataset of an energy community's generation and consumption with appliance allocation," *Data in Brief*, vol. 45, p. 108 590, 2022, ISSN: 2352-3409. DOI: `https://doi.org/10.1016/j.dib.2022.108590`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2352340922007971`.

[32] M. Khorasany, Y. Mishra, and G. Ledwich, "Design of auction-based approach for market clearing in peer-to-peer market platform," *The Journal of Engineering*, vol. 2019, no. 18, pp. 4813–4818, 2019.