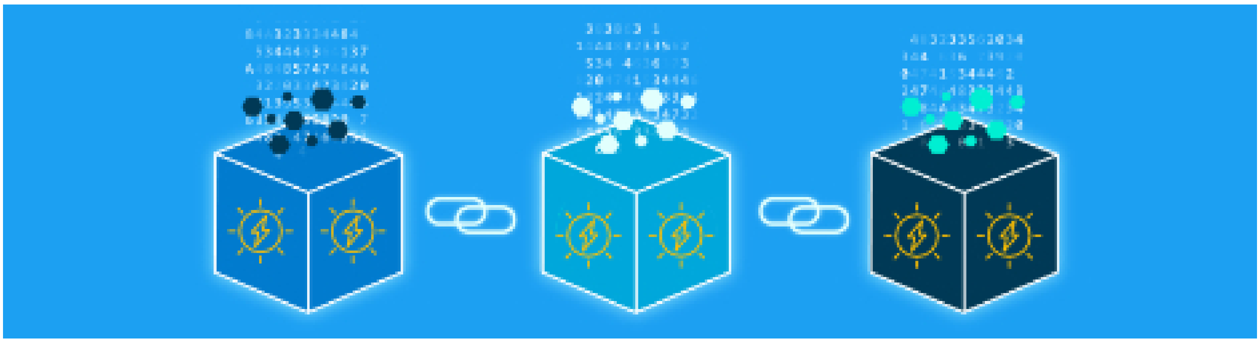


Power Share - Blockchain Management System and P2P Energy Trading

Rúben José Passos Martins

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Duarte Nuno Jardim Nunes
Dr. Sabrina Scuri

Examination Committee

Chairperson: Prof. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. Duarte Nuno Jardim Nunes
Member of the Committee: Prof. Hugo Gabriel Valente Morais

November 2019

Acknowledgments

I would like to thank my parents and close family members for their huge support in my studies, encouragement and caring over all these years, for always being there for me. Without them, this journey and thesis would have not been possible.

I would also like to thank my dissertation supervisors, Duarte Nuno Jardim Nunes and Sabrina Scuri, for their support and time that made this project possible.

My grateful thanks are also extended to the SMILE project team from M-ITI for their contribution to this project, especially to Eng. Daniel Garigali Pestana and Eng. Wilson Santos for all the help in the first parts of the project.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my IST student journey. Also, a special thanks to my colleague and friend Rodrigo Rocha from IST who accompanied me and helped during this thesis development journey and to Frank Ihringer who has been an awesome friend and partner.

Thank you

Abstract

Blockchain is a technology that could innovate the energy sector since it has the ability to manage, store and certify data with no need for an external service provider. Despite being costly to operate compared to a normal database system, the use of blockchain technology offers increased data transparency and immutability of records. Power Share combines the use of an Eco-Feedback system with Blockchain technology, allowing the users to sell and buy energy from their neighbours at better rates, instead of selling the energy to normal energy operators, challenging this way the current energy supply system and making the market more competitive. Connecting local prosumers and consumers also helps reduce the system's costs and improve grid stability through better management of the supply. The project described here will allow users to get control over their energy consumption, and decide from whom they want to buy energy and whether it should come from renewable sources, thus incentivizing the consumption of carbon-free energy. This project is implemented as part of SMart IsLand Energy systems (SMILE), a European Union (EU) H2020 funded research project, and will be pilot tested as a future business model for one of the solutions developed in the scope of SMILE. This solution consists on the implementation of a better performing and more efficient blockchain technology, based on Hyperledger Fabric, and the updating of the previously developed android app to work with the new environment. An admin interface was also developed to give additional functions to the possible blockchain administrator. The proposed solution is then critically analysed and evaluated by using both quantitative and qualitative approaches in order to understand if the new system is clear to end-users and assess its performance.

Keywords

Blockchain; Peer-to-Peer Energy Trading; Distributed Ledger Technologies; IBM Hyperledger; Renewable Energy

Resumo

A tecnologia blockchain pode vir a inovar o setor de energia, uma vez que tem a capacidade de gerir, armazenar e certificar dados sem a necessidade de um serviço externo. Em comparação com um sistema de dados normal, o uso da tecnologia blockchain, apesar de ter custos mais elevados, oferece uma maior transparência nos dados e imutabilidade nos seus registos. A Power Share combina o uso de um sistema de Eco-feedback com a tecnologia blockchain, permitindo aos utilizadores vender e comprar energia dos seus vizinhos a melhores preços, conseguindo assim desafiar o atual sistema de fornecimento de energia tornando o mercado mais competitivo. Esta conexão de consumidores e produtores de energia leva a uma redução de custos do sistema e melhora a estabilidade da rede elétrica. Este projeto irá permitir aos utilizadores controlar o seu consumo de energia, conseguindo decidir a quem comprar a energia e se provem de fontes renováveis, incentivando assim o consumo de energia limpa. Implementado como parte do SMILE, um projeto de investigação financiado pela EU H2020, o projeto aqui desenvolvido será testado como um futuro modelo de negócios para uma das soluções desenvolvidas no âmbito do SMILE. Esta solução consiste, assim, na implementação de uma mais eficiente tecnologia blockchain, baseada no Hyperledger Fabric, e na atualização do sistema atual. Uma interface administrativa foi também desenvolvida para fornecer funções adicionais ao possível administrador da blockchain. A solução proposta é analisada e avaliada criticamente usando abordagens quantitativas e qualitativas para entender se o novo sistema é claro para os utilizadores finais e de forma avaliar seu desempenho.

Palavras Chave

Blockchain; Transacções de Energia entre Pares; Tecnologia de Registo Distribuido; IBM Hyperledger; Energias Renováveis

Contents

List of Figures

List of Tables

Listings

Acronyms

ACL	Access Control Language
API	Application Program Interface
BEV	Battery Electric Vehicles
BFT	Byzantine Fault Tolerance
BP	Block Producers
CA	Certificate Authority
CPU	Central Processing Unit
D3A	Decentralised Autonomous Area Agent
DAG	Directed Acyclic Graph
DApps	Decentralized Applications
DApp	Decentralized App
DDOS	Distributed Denial of Service Attack
DLT	Distributed Ledger Technologies
DPoS	Delegated Proof of Stake
EEM	Empresa de Eletricidade da Madeira
EF	Eco-Feedback
ETH	Ethereum
ETMS	Energy Trading Management System
EU	European Union

EV	Electric Vehicle
EWF	Energy Web Foundation
HEV	Hybrid Electric Vehicles
HTTPS	HyperText Transport Protocol Secure
IoT	Internet of Things
LTS	Long Term Support
MVVM	Model View - View Model
P2P	Peer–to–Peer
PBFT	Practical Byzantine Fault Tolerance
PHEV	Plug-in Hybrid Electric Vehicle
PSv1	Power Share version 1
PSv2	Power Share version 2
PV	Photovoltaic
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
PoeT	Proof of Elapsed Time
RAM	Random Access Memory
RES	Renewable Energy Sources
SMILE	SMart IsLand Energy systems
SSH	Secure Shell
SSL	Secure Sockets Layer
TDP	Thermal Design Power
TLS	Transport Layer Security
TSO	Transmission System Operator

UI	User Interface
VPS	Virtual Private Server

1

Introduction

Contents

In the 21st century, the growing increase in energy consumption and the constant emission of polluting gases are still problems to be solved. Clean and renewable energy sources are increasingly being used to replace fossil fuels, nevertheless, they are not enough to meet the world's energy demand. Thus, increasing efficiency and reducing energy waste is more important than ever before.

To face the current energy crisis and reduce the environmental impact of energy production and consumption, both prosumers - "energy consumers who also produce their own energy from a range of different onsite generators; mainly from renewable energy sources." [1] - and consumers are being asked to play an increasingly active role in the management and monitoring of their consumption. Several are the solutions developed specifically to support people in performing this task and, among them, it is worth mentioning the eco-feedback technology, which aims to provide "feedback on individual or group behaviours with a goal of reducing environmental impact [2], and Distributed Ledger Technologies (DLT). In addition, DLT, and especially blockchain, are gaining increasing attention into the energy sector, where their most promising applications seem to be Peer–to–Peer (P2P) energy trading and energy supply certification. Thanks to such technologies, energy consumers and prosumers can now buy and sell energy between each other. For prosumers, this means having an alternative to selling the excedent energy to the electricity company at a very low fixed-rate (feed-in tariff). While consumers that don't have their own energy generation system could buy local, green energy, at a lower rate, directly from those producers that are available to sell their excedent energy.

This trustless DLT energy trading - where "trustless" means that the amount of trust required from any single actor in the system is minimized and distributed among different actors in the system [3] - besides allowing small consumers and prosumers to finally enter the energy market, has the advantage of reducing energy losses in distribution since it encourages the energy to be produced and consumed locally, decreasing the energy loss due to transmission over longer distances [4].

It is not only prosumers and consumers that could benefit from the integration of blockchain in smart grids but also Transmission System Operator (TSO), as using blockchains together with smart meters installed in each household would make billing and metering processes faster and automated. [4] The implementation of blockchain technology would also allow to track the energy consumed from where it was generated, providing operators precise data on the energy sources in the grid, thus increasing reliability and resilience of the local power supply. These benefits would result in significant reductions in maintenance costs, while end-users, both prosumers and consumers, will be the most benefited from a more efficient system. Despite their clear potential, Blockchain and DLT are still "new" technologies, thus a lot of work still needs to be done in order to get a better understanding of their application to P2P energy trading.

The work described here is based on a previous master thesis [5] where a P2P energy trading platform combining Eco Feedback Eco-Feedback (EF) and DLT - Power Share - was implemented to simulate the energy exchange within a small neighbourhood community in Funchal (Madeira Island). The goal of this project is to solve the main issues emerged from the deployment of Power Share and thus develop a more reliable and user-friendly P2P energy trading platform.

The major changes made concern the back-end side of the system, with the final goal of improving system performance and efficiency. Overall, the main focus was on the development of a new transaction system, released together with a new admin interface, providing a simpler payment system for energy consumers and prosumers. Minor changes on the front-end side of the end-user android app were also made in order to integrate the new backend system.

1.1 The SMILE Project

This project, as the previous one, is integrated into the H2020 SMart IsLand Energy systems (SMILE) project which aims to test and demonstrate some smart grid technologies, as well as business models, within large scale projects (`https://smile.m-iti.org/project-summary`).

SMILE consortium is composed of 19 partners from 6 European Union (EU) countries and is integrated into 2030 Energy Strategy agreement, which aims to achieve energy savings and increase the share of renewable energy in order to create a more competitive, secure and sustainable energy system between 2020 and 2030. Currently, SMILE involves three large scale pilot projects in three European Islands (Madeira in Portugal, Samsø in Denmark and Orkneys in the United Kingdom) with similar topographic characteristics but different policies, energy markets and regulations.

The choice of having large-scale pilot projects being tested on island locations provides a fundamental advantage to the research where these remote communities can be more easily engaged in the real-life testing of solutions aimed at solving important energy challenges.

Madeira is an interesting case since it's a total energy island, whose energy network has no connections to the mainland. In the scope of SMILE, Madeira demonstrator is running three different pilots aimed respectively at (a) optimizing self-consumption of solar energy; (b) implementing a low-cost smart charging solution for Electric Vehicle (EV)s; and (c) providing frequency and voltage control mechanisms.

The first version of the Power Share app, as well as the project described here, are designed within the scope of the first pilot. More detailed information regarding SMILE and the current pilots is available at `http://www.h2020smile.eu`.

1.2 Objectives

The peer-to-peer network developed in the scope of the previous master thesis [5] was deployed in Madeira Island and used to simulate the energy trading within a small neighbourhood community. Payments were performed using the IOTA network and energy was exchanged by means of its cryptocurrency (mIOTA). That choice was due to the fact that, when the project was conceived, IOTA appeared to be one of the most promising DLT for P2P energy trading. This, not only because it's cryptocurrency was one of the top 10 by market cap according to coinmarketcap website (`https://coinmarketcap.com`), but especially because, unlike blockchain, IOTA is based on a different data structure: the Tangle, a Directed Acyclic Graph (DAG) that is blockless. Compared to blockchain, this protocol is theoretically lightweight, more scalable, and has no transaction fees. For these reasons, IOTA is often presented in literature as a valuable alternative to blockchain for P2P energy trading, nonetheless, was never empirically tested before. Despite the theoretical advantages of IOTA over blockchain, results from the previous study [5] show several limitations of such technology, leading to conclude that, in its current state, IOTA is not suitable for P2P energy trading. Indeed, at the end of the deployment, the server that managed the payment network had a very high energy consumption, greater than the energy exchanged between participants in the study. Also, transaction speed was much lower than expected, affecting the user experience of the trading system.

The main objective of this research project is to develop a blockchain-based, improved version of the previous system [5]. Major effort will be put on the enhancement of the system performance in terms of transaction speed and payment management. With better integration of blockchain technology, the resulting system will provide a reliable and fast decentralized mechanism to exchange electrical energy, allowing users to choose from where they want to buy electrical energy.

1.3 Proposed Approach

The expected result of this work will be the release of a new version of the android app developed for the scope of the previous master thesis, improved in terms of app performance and stability but especially with an integration with the new blockchain system. Some refinements to the interface and source code of the trading process will also be made.

The app server that provides the correct REST Application Program Interface (API) to the android app will be reviewed and adapted to work seamlessly with the new blockchain system. The blockchain system will have an admin interface where the blockchain operator or the Power Share admin could test the network transactions and check all the assets and participants in the network. By using a distributed ledger technology that is different from that used in the previous version of Power Share (from now on called Power Share version 1 (PSv1)), we aim to develop a system which is less energy demanding, faster and more transparent to the end-users.

To evaluate the proposed solution, a second pilot will be run in Madeira involving the same sample that took part in the field-test of Power Share. At the end of the study, results will be compared with those from the deployment of PSv1, and conclusions will be elaborated.

1.4 Document Outline

The present document is structured as follows: chapter 2 provides an overview of related work, the current state of the art and some concepts regarding the main DLT currently being used in the energy sector. The proposed solution, the technologies that suit best the project, as well as the different specifications for each target and the detailed work done on the PSv1 system is presented in chapter 3. Evaluation methodology and preliminary results are described in chapter 4, while conclusions are elaborated in chapter 5.

2

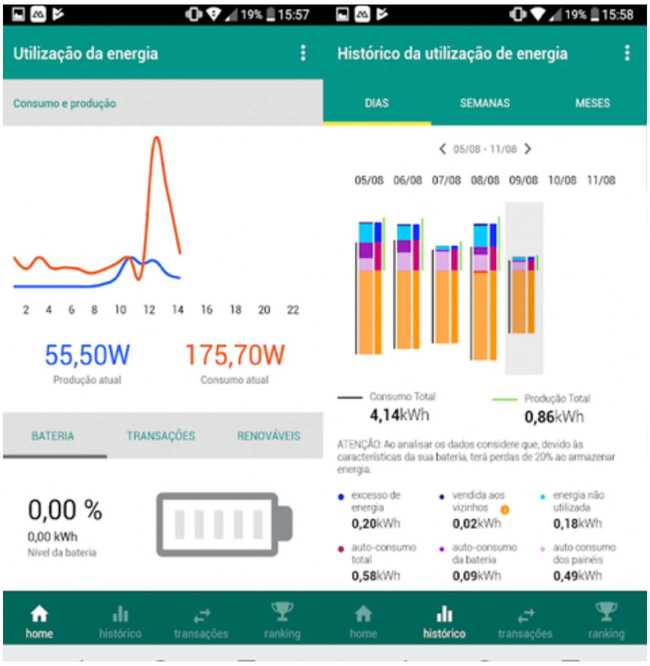
Related work and Theoretical Background

Contents

This chapter presents some background and related work on blockchain technologies and its applications in the energy sector. The chapter is divided into three main sections,(a) 2.1 Power Share version 1, where the previous work done on the Power Share system is described; (b) 2.2 Blockchain, were important concepts about this technology are presented; and (c) 2.3 Blockchain Technology in the Energy Sector, which covers some examples of the applications of blockchain in the energy market.

2.1 Power Share Version 1

Power Share is a P2P energy trading system resulting from the previous master thesis [5]. It mainly consists of an Android application connected with an Energy Trading Management System, that performs payment via IOTA and contemporary provides EF using production and consumption data collected through the smart meters installed in the houses of participants in the project. Data from the smart meters are collected in a server that is also responsible for managing energy offer and demand on the peer-to-peer network, providing users with feedback on their energy consumption and production data and managing their account settings. The communication between the android app and the server is made via a REST API.

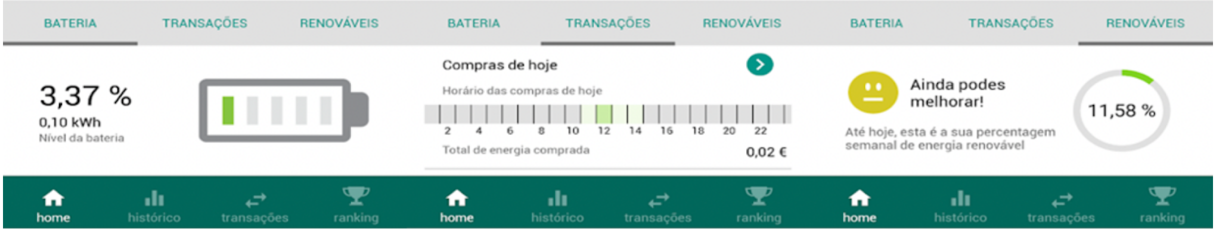


ATENÇÃO: Ao analisar os dados considere que, devido às características da sua bateria, terá perdas de 20% ao armazenar energia.

Figure 2.1: PSv1 home screen and energy consumption and production status

2.1.1 The End-user Application

In the main section “Utilização de Energia” (represented in figure 2.1), Power Share app provides users with real-time feedback on energy consumption and production, which is often defined as a requirement by users of eco feedback systems [5] since it helps them get a better understanding of their energy usage patterns. PSv1 also provides a set of information related to excess energy stored in the prosumer’s battery (however, since none of the participants in the study owns a battery, this data was based on simulations), the current day’s transactions and the percentage of renewable energy consumed (represented in figure 2.2).



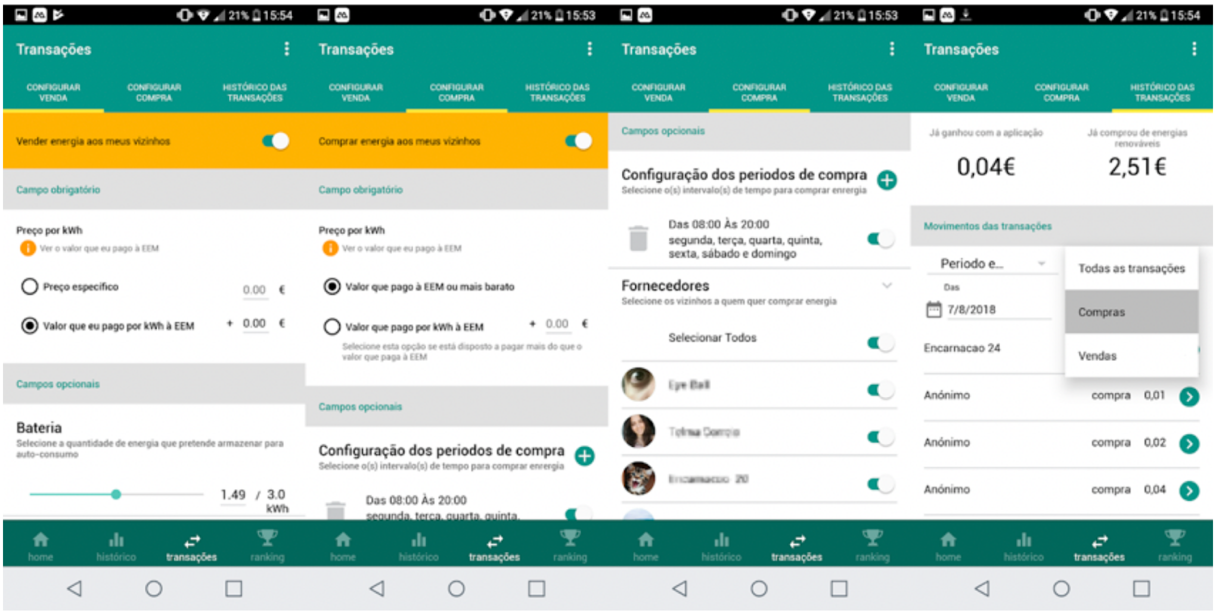
3,37 %
0,10 kWh
Nível da bateria

Compras de hoje
Horário das compras de hoje
Total de energia comprada 0,02 €

Ainda podes melhorar!
Até hoje, esta é a sua percentagem semanal de energia renovável
11,58 %

Figure 2.2: PSv1 home screen tabs

Besides real-time feedback, PSv1 also provides historical feedback, that is to say, it allows users to access their energy consumption and production data over a defined period of time (in this case, on a daily, weekly and/or monthly basis). During the field test of PSv1, historical feedback was one of the most appreciated features, since it provided some useful data to better define the best criteria to buy or sell energy [5].



Vender energia aos meus vizinhos
Campo obrigatório
Preço por kWh
Ver o valor que eu pago à EEM
○ Preço especifico 0.00 €
● Valor que eu pago por kWh à EEM + 0.00 €
Campos opcionais
Bateria
Selecione a quantidade de energia que pretende armazenar para auto-consumo
1.49 / 3.0 kWh

Comprar energia aos meus vizinhos
Campo obrigatório
Preço por kWh
Ver o valor que eu pago à EEM
● Valor que pago à EEM ou mais barato
○ Valor que pago por kWh à EEM + 0.00 €
Selecione esta opção se está disposto a pagar mais do que o valor que paga à EEM
Campos opcionais
Configuração dos periodos de compra
Selecione o(s) intervalo(s) de tempo para comprar enrergia
Das 08:00 Às 20:00
segunda, terça, quarta, quinta,

Campos opcionais
Configuração dos periodos de compra
Selecione o(s) intervalo(s) de tempo para comprar enrergia
Das 08:00 Às 20:00
segunda, terça, quarta, quinta, sexta, sábado e domingo
Fornecedores
Selecione os vizinhos a quem quer comprar energia
Selecionar Todos

Já ganhou com a aplicação 0,04€
Já comprou de energias renováveis 2,51€
Movimentos das transações
Periodo e...
Das 7/8/2018
Encarnacao 24
Todas as transações
Compras
Vendas
Anónimo compra 0,01
Anónimo compra 0,02
Anónimo compra 0,04

Figure 2.3: PSv1 transaction feature screens

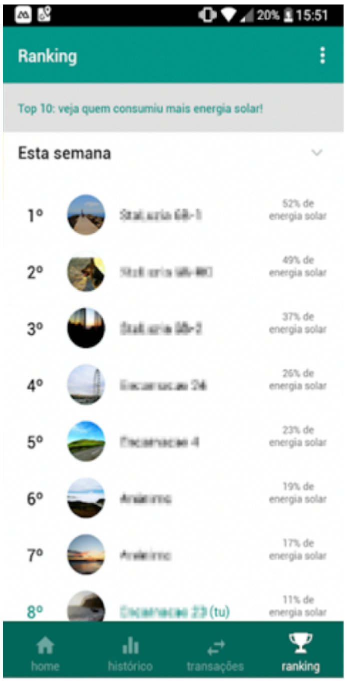
A further feature of the app, the trading section, is divided in 3 different sub-sections (represented in figure 2.3):

- manual settings for selling energy;
- manual settings for purchasing energy;
- list of all transactions performed by the user;

Through trading section, users have the opportunity to manually define criteria for buying and/or selling energy. Price per kWh is the only mandatory criteria to set. In this case, there are two possible options:(1) a fixed price defined by the user or (2) a price tied to the cost of energy bought from the electricity company.

In addition, users can select people they want to trade with, among all users in the Power Share network, as well as define specific time frames for selling and buying energy, and/or set a specific amount (in kWh) of the overall battery capacity they want to save for self-consumption only.

At any time, users can disable and enable the trading, modify trading settings, and switch from manual to automatic settings (or vice versa).



Ranking

Top 10: veja quem consumiu mais energia solar!

Esta semana

1º [redacted] 52% de energia solar

2º [redacted] 49% de energia solar

3º [redacted] 37% de energia solar

4º [redacted] 26% de energia solar

5º [redacted] 23% de energia solar

6º [redacted] 19% de energia solar

7º [redacted] 17% de energia solar

8º [redacted] (tu) 11% de energia solar

Figure 2.4: PSv1 ranking screen

“Ranking” is the last section of the app (represented in figure 2.4), which presents the top ten users that, each week, have consumed (in percentage) the most energy from renewable sources.

2.1.2 The System Back-end

Alongside the android app, a java with Spring framework web server was created to handle the API requests from the android app and receive the readings of energy production and consumption from the smart meters. This system, called Energy Trading Management System (ETMS), featured a MySql database to store all the user information, trading criteria and energy readings, and an authentication scheme with Bearer Authentication [5]. The energy trading algorithm in the ETMS was another important feature of this system and consists of three main asynchronous tasks:

- **Start Transactions:** Where all the users that want to sell and/or buy energy are analyzed and, if a match between a seller and buyer settings is found, a temporary transaction is created between the two users;
- **Verify Transactions:** Every two minutes the system analyses the list of temporary transactions and verifies if the energy exchanged is greater than zero. If after 5 minutes the amount of energy exchanged is still zero, the transaction is stopped and both buyer and seller entries are updated into the database;
- **Stop Transactions:** The ETMS will be checking the temporary transactions and, after an hour of successful energy trading, these transactions are terminated. A final transaction is then created in the database with the total energy exchanged during the one hour period. Both buyer and seller data are updated and available to start a new transaction.

For the purpose of PSv1 project, IOTA cryptocurrency - mIOTA, a tangle based cryptocurrency that aims to provide a safe and decentralized financial ecosystem for the Internet of Things (IoT) [6] - was used as a mean of exchange for energy. Payments were performed each time a final energy transaction is made on the ETMS.

Despite the theoretical advantages of IOTA over blockchain [7], the main benefits that blockchain technologies could bring - like the automation of transactions using Decentralized Applications (DApps) and smart contracts (described in the following pages) - could not be implemented. Moreover, at the end of the deployment, several drawbacks of using IOTA technology emerged. Among them the system power consumption. Indeed, in order to validate the transactions, the Virtual Private Server (VPS) - running on an Ubuntu 16.04.4 server equipped with a 6 cores processor and 8GB Random Access Memory (RAM) [5] - consumed more energy than the amount exchanged within the community during a one-month deployment. Also, further issues related to the addresses' generation process and the unexpected length of the transactions' validation process (i.e. synchronization and latency of transaction propagation), that negatively affected the user experience.

2.2 Blockchain Technology

2.2.1 Introduction

Blockchain is a secure and transparent distributed ledger technology used for the transmission and storage of data. It relies on a decentralized system, a peer-to-peer network of interconnected computers, which does not require a central supervising authority. This technology was first proposed in 2008 by Satoshi Nakamoto (pseudonym of the person or group of people who created Bitcoin), that published the Bitcoin white paper [8] describing the first digital currency developed to allow peer-to-peer payments and solve the double-spending problem in a decentralized fashion without the need of a central trusted party.

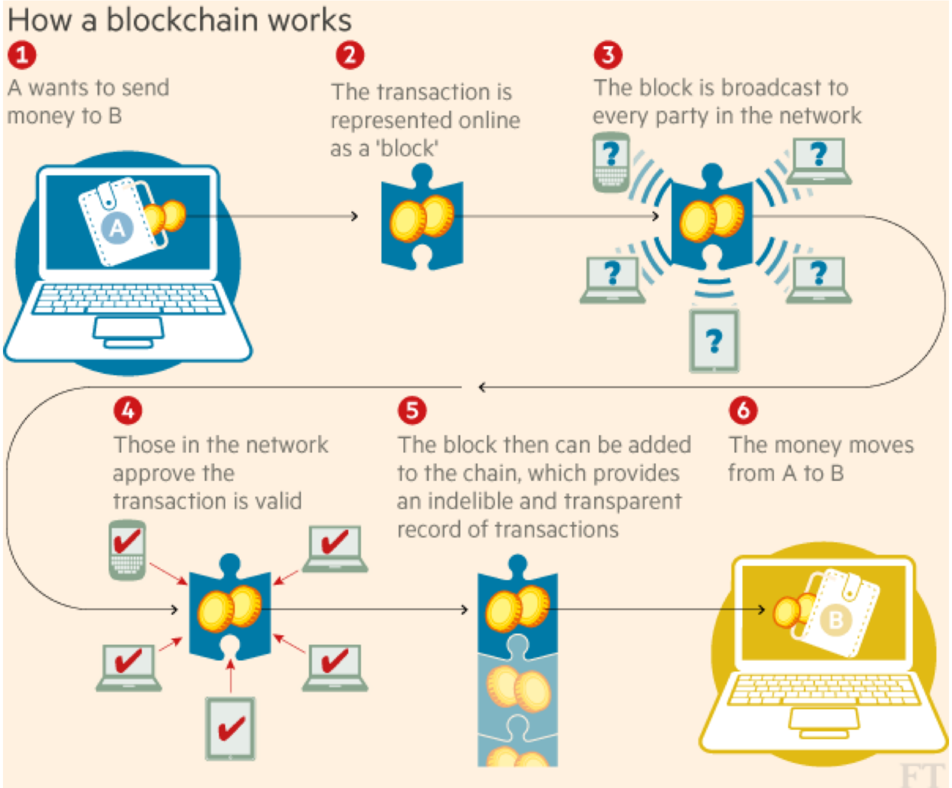

How a blockchain works
1
A wants to send money to B
A
2
The transaction is represented online as a 'block'
3
The block is broadcast to every party in the network
?
4
Those in the network approve the transaction is valid
5
The block then can be added to the chain, which provides an indelible and transparent record of transactions
6
The money moves from A to B
B
FT

Figure 2.5: Representation of transactions process within a blockchain system

With blockchain, transaction data are not stored on a centralized server, instead, they are distributed across a P2P network of computers (see figure 2.5), that's why we refer to blockchain as a Distributed Ledger Technology.

Blockchain enables trustless networks, where 'trustless' means that all the parties could make transactions without the need to trust each other. This is possible thanks to the use of cryptography, which brings authoritativeness behind all the interactions in the network. This use of cryptography, however, doesn't completely eliminate trust in the system but it minimizes the amount of trust required from any single participant by running an economic game that incentivizes participants to cooperate with the rules defined by the blockchain protocol. Blockchain applications also increase the trustworthiness and data provenance with "immutable properties that can be distributed and validated independently by any entity across boundaries or authority enclaves" [9].

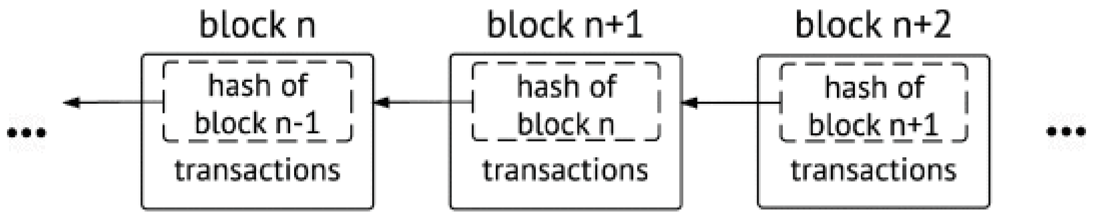


Figure 2.6: Representation of the chain of where each block has the hash of the previous block and the list transactions made.

Each block (see figure 2.6) presented in the blockchain is identified by a hash, generated using the SHA256 cryptographic hash on the header of the block. It contains a package of transactions, an associated timestamp, a nonce and the hash of its n-1 block inside its own header, creating in this way a sequence of blocks that will go back to the first block created, the genesis block [10]. This genesis block is special because it has no parent block, containing any hash of a previous block because it's the first block on the blockchain.

2.2.2 Permissioned and Permissionless Blockchains

Blockchains can also have different access restrictions, which distinguish between Permissionless blockchains and Permissioned blockchains [11].

Permissionless Blockchains

Permissionless Blockchains, also known as public blockchains, don't require any authorization to, read and write data on the blockchain. In other words, there isn't any kind of censorship on permissionless blockchains, any party in the network can read from the ledger and/or create a wallet and make transactions [8, 12]. Moreover, any user can run a validation node and purchase mining rights to support the blockchain operations. In these networks, validators are normally rewarded for their contribution. Bitcoin and Ethereum are the most famous networks that use a public blockchain.

Permissioned Blockchains

In permissioned Blockchains, also known as private blockchains, users require some sort of permission or authorization to access some (if not all) parts of the blockchain and only trusted nodes are allowed to verify the transactions [12, 13]. These kinds of blockchains are normally used by companies to securely record transactions or exchange data between each other.

Hyperledger is one of the most used blockchains that could work as a private blockchain.

There are some variants of permissioned blockchains, based on the amount of write and read rights given to users. Ripple, as an example, is a semi-permissioned blockchain, where the validator nodes are owned by Ripple labs but any user could read and send transactions on their blockchain network [14].

Permissioned blockchains are considered more strict and controlled systems since all the participants in the network are whitelisted and bounded by strict contractual obligations to behave “correctly”. In addition, more efficient consensus protocols are used in these networks, such as Practical Byzantine Fault Tolerance (PBFT) [15] used in Hyperledger Fabric.

2.2.3 Consensus Algorithms

Consensus algorithms are the mechanism that provides security in a blockchain by ensuring that no malicious transactions or changes can be made on the blockchain, that is to say, guaranteeing “that the nodes agree on a unique order in which entries are appended” [12]. One of the major blockchain problems is the double spending problem which is a “potential flaw in a cryptocurrency or other digital cash scheme whereby the same single digital token can be spent more than once, and this is possible because a digital token consists of a digital file that can be duplicated or falsified ” [16]. Bitcoin was the first blockchain to propose a solution to this problem with a Proof of Work (PoW) consensus algorithm.

A brief description of the major validation mechanisms on permissionless blockchains (PoW, Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) consensus algorithms) will be made. Also, on permissioned blockchains, which are more widely used in an enterprise context where participants are known, PBFT and Proof of Authority (PoA) consensus (that are more centralized and efficient consensus validation mechanisms compared to the ones used permissionless blockchains) will be detailed.

Proof of Work

PoW is used in the bitcoin [8] where network participants (also known as miners [17]) compete with each other to add the next transaction block to a blockchain by solving a complex cryptographic puzzle, which leads them to validate prior transactions in the process and earning transaction fees for their work. This consensus protocol is probably the most widely used in the blockchain ecosystem. In 2016 PoW powered blockchains accounted for more than 90% of the total market capitalization of digital cryptocurrencies [18]. Nonetheless, PoW validation process is highly demanding in terms of energy and computational power. Besides, the complexity of verifying the hash functions is also increasing over time. This leads to very high energy consumption and a big ecological footprint.

Proof of Stake

With Proof of Stake PoS, the validation process is different. Network validators are connected to each other with servers (also known as masternodes) and they invest in tokens or cryptocurrencies in the blockchain network, representing their stake in the block. PoS attempts to solve the energy expenditure problem created by PoW. To do so, PoS replaces PoW's competition by pseudo-randomly selecting stakeholders to append to the blockchain [19]. Then, the validator chance of verifying a block is proportional to his stake in the block so, for example, if you have 15% of the total circulating supply, you have a 15% chance to be chosen as a validator of the block and thus receive the respective transaction fees. Compared to the PoW mechanism, PoS system is by design less energy-intensive because there's no mining involved, requiring much less computer power to validate a transaction [20].

Delegated Proof of Stake

There is also a variation of PoS: the DPoS, which uses real-time voting combined with a social system of reputation to achieve consensus. In DPoS every token holder can vote for the blockchain main elected delegates. This protocol was introduced in 2014 by Dan Larimer on the BitShares cryptocurrency [21], and has been applied to other well-known cryptocurrencies like Steem, EOS, and Lisk. In this protocol, block producers can collaborate to add blocks in the blockchain instead of competing like in PoW and PoS. There are generally between 21 to 100 elected delegates in a DPoS system and they could be kicked out of the system if they keep failing to provide their contribution in validating the blocks (ex: offline nodes or nodes with a malfunction), so new nodes can take their position on the blockchain system. DPoS is considered to be a more centralized blockchain consensus protocol, however, it's more efficient and it could reach much faster block times (less than a second in case of the EOS blockchain [22]) than the other PoW or PoS mechanisms.

Practical Byzantine Fault Tolerance

In distributed computing, a group of nodes communicate with each other in order to reach consensus. A common problem in such scenario is when some of the nodes have a so-called “Byzantine failure”, i.e. provide malicious or conflicting information. This issue is described in the Byzantine Generals Problem [23]. Based on signed messages and partially synchronous network, the PBFT consensus algorithm is capable of handling malfunctioning nodes and generating consensus in one of those environments. [24]

PBFT is used on the IBM Hyperledger Fabric, Ripple and Stellar blockchains - where the number of validation nodes is limited and their identity is known (since they are part of a permissioned blockchain network) - and provides a low-cost network with a high transaction throughput.

Proof of Authority

PoA is a consensus algorithm with better performance than Byzantine Fault Tolerance (BFT) algorithms resulting from lighter message exchanges [25]. With this algorithm, transactions are validated by pre-selected authorities (named validators). These nodes are the authority of the blockchain system and new blocks are created only when the majority of them reach a consensus.

The PoA creates a very centralized system but it is easily scalable and allows a very high transaction throughput. Although it’s not being widely used, PoA is implemented on Ethereum testnets (Rinkeby and Kovan) and being used by newer blockchain startups like Poa.Network.

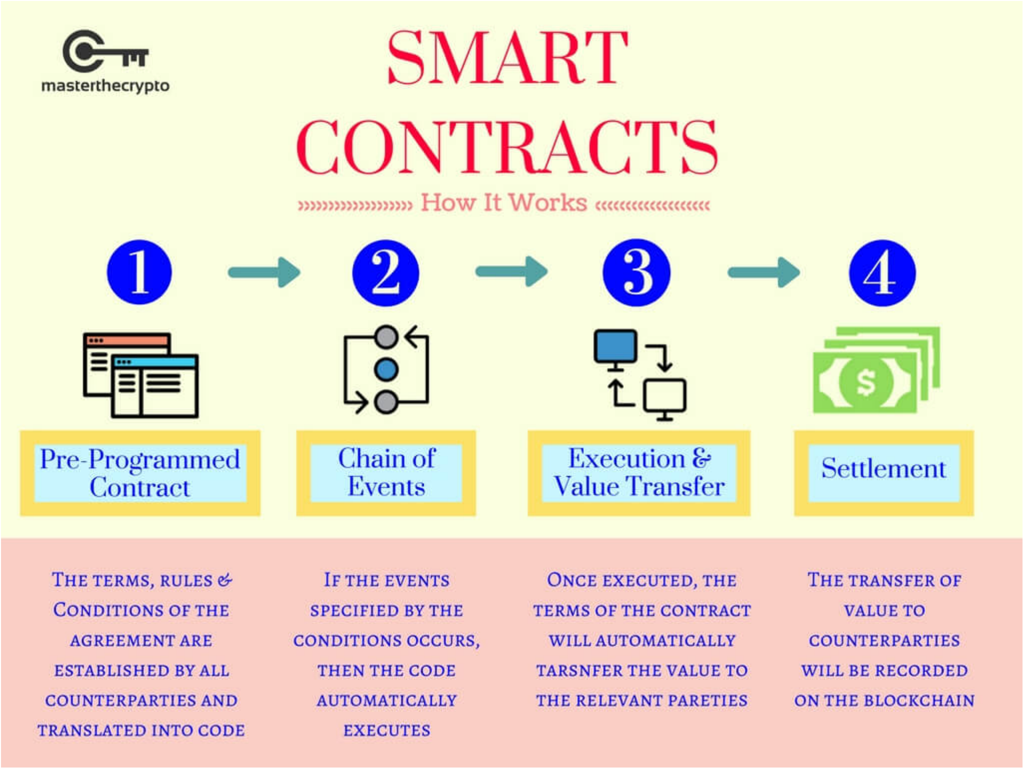
2.2.4 Major Smart Contract Blockchain Projects

In the context of our research, the most well-known and widely used blockchains for smart contract development are Hyperledger Fabric, Ethereum and EOS. These three networks were identified as the most suitable option for the development of Power Share system, so it is appropriate to dive deep into them.

2.2.4.A Ethereum

Ethereum is an open-source blockchain, created by Vitalik Buterin, which allows building DApps - “which is a novel form of the blockchain-empowered software system” [26] that runs on smart contracts and could have a frontend user interface - and run smart contracts on a public blockchain. [27]. Ethereum currently uses PoW as it’s consensus algorithm, where miners get rewarded for their computational effort to run the blockchain, and could be known as a virtual machine (Ethereum Virtual Machine), designed to remotely supply all the services of a computer, like a distributed cloud. This technology uses its own virtual currency called Ether, which is the most used virtual currency after Bitcoin.

The revolution introduced by Ethereum was the establishment of smart contracts. Smart contracts are self-executing scripts that run without the need of any intermediaries. Smart contracts reside on the blockchain, define the rules related to the transaction - like a protocol that executes the terms of a contract [28] - and allows proper, distributed and automated workflows (see figure 2.7).



THE TERMS, RULES & CONDITIONS OF THE AGREEMENT ARE ESTABLISHED BY ALL COUNTERPARTIES AND TRANSLATED INTO CODE

IF THE EVENTS SPECIFIED BY THE CONDITIONS OCCURS, THEN THE CODE AUTOMATICALLY EXECUTES

ONCE EXECUTED, THE TERMS OF THE CONTRACT WILL AUTOMATICALLY TARSNFER THE VALUE TO THE RELEVANT PARETIES

THE TRANSFER OF VALUE TO COUNTERPARTIES WILL BE RECORDED ON THE BLOCKCHAIN

Figure 2.7: Representation of how a smart contract works

Smart contracts are also used in DApps in order to create applications that interact with the blockchain. They operate autonomously, with no need of an authority controlling the tokens issued, and store data on a public, decentralized ledger. DApps also reward their users for the validation process of new insertions in their public decentralized ledgers with tokens of their cryptocurrencies. Some examples of DApps on Ethereum are, Cryptokitties, Basic Attention Token, and Golem.

In short, Ethereum is characterized by the following main features:

- It's a public blockchain where everyone could access his/her data;
- It allows users to create smart contracts and run DApps that interact with the blockchain composed of thousands of decentralized computers connected;
- It uses proof of work as a consensus algorithm but has planned to switch to proof of stake in 2020.

2.2.4.B EOS

EOS is a DPoS consensus-based blockchain system that allows the developers of an Decentralized App (DApp) to have separate authentication systems, databases, and a system of communication and inter-connectivity with other DApp's [22]. The resulting technology could scale to millions of transactions per second, thus allowing for quick and easy deployment of DApp's, and eliminates transaction fees.

There are 3 main parties that interact with each other in the EOS network (see figure 2.8):

- Block Producers (BP)'s, generally between 21 to 100 elected delegates, that are shuffled periodically to validate the blockchain transactions. If BP misbehave (publish invalid transactions) or go offline the EOS community can vote them out and replace them with better BP. EOS is an inflationary cryptocurrency and blockchain BP's get EOS coins as a reward from their computational power;
- EOS users, which (a) have access to a wallet storing EOS tokens and cryptocurrency, (b) can vote to elect BP's and (c) can interact with DApps or send EOS currency to each other;
- DApps which, like in Ethereum, are applications that are created in a decentralized blockchain environment. In the EOS case, DApps are required to stake EOS coins in order to get computing resources (Central Processing Unit (CPU) time, RAM space and network traffic) from BP's.

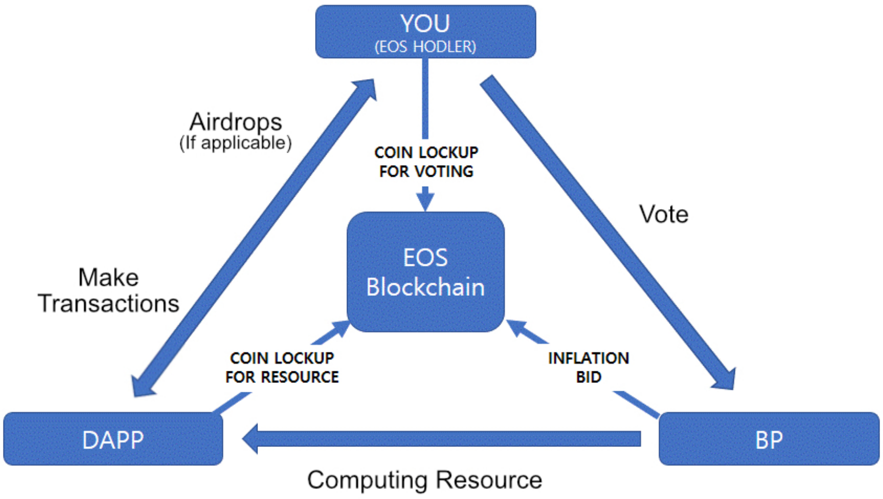


Figure 2.8: EOS value circulation mechanism

In short, EOS is characterized by the following main features:

- The blockchain used is public, it allows the deployment of smart contracts and cryptocurrency transactions don't have any fees;

- It is one of the best blockchains in terms of performance, with block times that are less than one second;
- It uses DApp as consensus algorithm, which is more environmentally friendly due to the reduced energy consumption compared to the PoW and PoS consensus algorithms.

2.2.4.C HyperLedger Fabric

Hyperledger Fabric is a distributed operating system for permissioned blockchains. This modular and open-source system is developed by Hyperledger, led by the Linux Foundation in partnership with some well-known companies, like Intel, IBM, JP Morgan, and Cisco. [29]

Hyperledger aims to reinforce the use of blockchain technology in different sectors, like supply-chain management, IoT, and finance, with the final goal of standardizing the use of blockchain in the business sector and increasing its adoption in commercial use. Unlike the previous blockchain technologies that were presented, HyperLedger Fabric doesn't reward it's nodes to maintain the blockchain running and doesn't use the most known blockchain consensus algorithms like PoW or PoS. Since it's a permissioned blockchain and its participants are known and identified, Hyperledger Fabric can use traditional BFT consensus "providing a way to secure the interactions among a group of entities that have a common goal but do not fully trust each other, such as businesses that exchange funds, goods, or information" [29].

Hyperledger features 3 types of nodes (see figure 2.9):

- **Clients:** which act like a blockchain end-user that communicate with orderers and peers and could invoke blockchain transaction or read data from it if the necessary permissions are granted ;
- **Peers:** that are the blockchain nodes that maintain the ledger, participate in consensus protocol by communicating with each other and then commit new transactions to the ledger;
- **Orderers:** the nodes that collectively form the ordering service, and provide the communication channels that peers and clients use to broadcast messages containing transaction details.

The communication channels mentioned above are also a feature that distinguishes Hyperledger from other blockchain technologies, where clients could only see the blockchain messages on the channels that they are authorized to access and the access to transactions is restricted to the involved parties.

Hyperledger, alongside the Hyperledger Fabric development, also produces and supports a wide range of blockchain technologies and frameworks that aim to make an easier and efficient development in the blockchain environment [30].

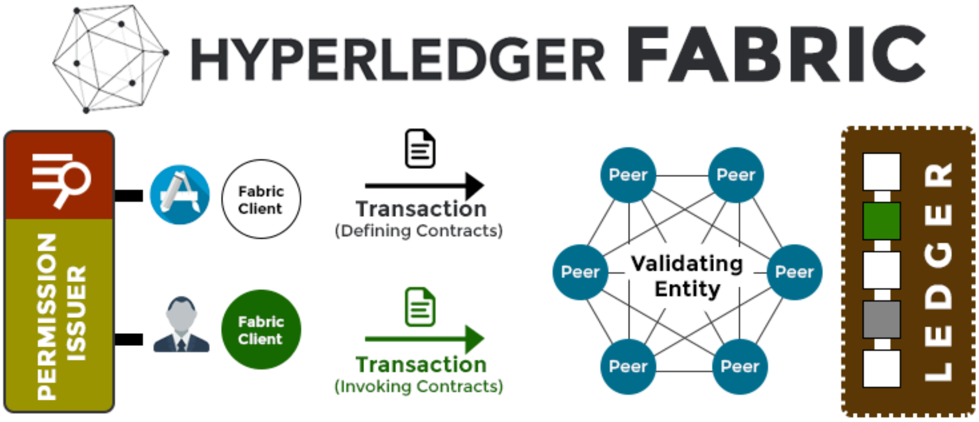


Figure 2.9: Representation of Hyperledger Fabric System

Some projects help the developers, that work with Hyperledger Fabric, to create business networks more efficiently (Hyperledger Composer), to create user-friendly blockchain admin panels to test the blockchain, query blocks, as well as check chaincodes and transactions or associated data (Hyperledger Explorer). Some other special blockchains can be found inside the Hyperledger environment, like Hyperledger Burrow, which works along with the Ethereum blockchain and Hyperledger Sawtooth (developed by Intel and using Proof of Elapsed Time (PoeT) as consensus algorithm). Finally, some blockchain deployment services are also available within the Hyperledger environment, among them Hyperledger Cello, which offers a blockchain-as-a-service deployment.

In summary, Hyperledger Fabric is characterized by the following main features:

- The use of a private blockchain, where interested parties will require access right to interact with the network. This characteristic distinguishes Hyperledger from other systems like Bitcoin or Ethereum, which are based on public blockchains where anyone could make transactions and read all the data from the ledger.

- There is no mining and no algorithms are required to secure the transactions, so by relying on the identities of the connected peers, PBFT consensus is used [29].

- Transactions in the network could be confidential, visible only to those that have the correct encryption keys.

- Smart contracts could be used to run code in the blockchain in order to automate business processes.

2.3 Blockchain Technology in the Energy Sector

Blockchain is not yet a mature technology, thus it still needs to be fully adopted and understood. Nowadays, the most advanced applications of this technology can be found in the finance industry, nevertheless, its disruptive potential is finally being explored also in the energy sector.

This technology has the potential to disrupt the current energy market by creating a new decentralized, democratized, transparent and more efficient energy supply systems. Nowadays, the energy market is a monopoly controlled by the main distribution companies and there is no transparency in the way they operate. Energy prosumers have difficulties selling their surplus energy at a fair price because most of the times it can be sold only to the electricity company at a fixed (and usually quite low) tariff, while consumers can only buy energy from the grid operator, and without any guarantees that it comes from a renewable energy source.

Blockchain emerges as a possible workaround for these issues since it provides a secure, decentralized, P2P network where smart contracts would automate transactions, thus ensuring information integrity, transparency, and confidence between all parties [31]. In addition, payments could be made via cryptocurrencies, leading to faster payments and, most importantly, allowing users to make micropayments (which are common in P2P energy trading) for virtually zero overhead costs since there will be no intermediaries to handle the transactions.

Despite P2P energy trading is the most common application of blockchain in the energy sector, other use cases are being explored, among them green energy supply certification [32], services for providing grid stability through energy redispatch [33], and fairness-control within micro-grids [34]. Examples of existing projects are provided below.

Power-ID

Power-ID [35] is a pilot project led by ETH Zurich that is being tested in the village of Walenstadt. The objective is to create a small, local P2P energy marketplace that leverages blockchain technology to make network costs completely transparent. The system focuses on solar energy and battery storage. It aims to cover at least half of Walenstadt's energy needs by connecting local players, so as to reduce the system's costs, and encourage renewable energy production and consumption. In this pilot project, the cost of the network is determined in a bottom-up way instead of being imposed by the large network operators.

NGRCOIN

Created by researchers from the Vrije Universitet Brussel. The idea behind the NRGCoin project is to encourage energy consumers to use local green energy by paying prosumers in a cryptocurrency called NRGcoin [36]. It aims to increase the value of prosumers renewable energy installations and reduce the cost of using local produced green energy. It uses Ethereum smart contracts to automatically mine new NRGcoins for each kWh of renewable energy injected into the grid, as well as to manage energy purchases and sales (see figure 2.10).

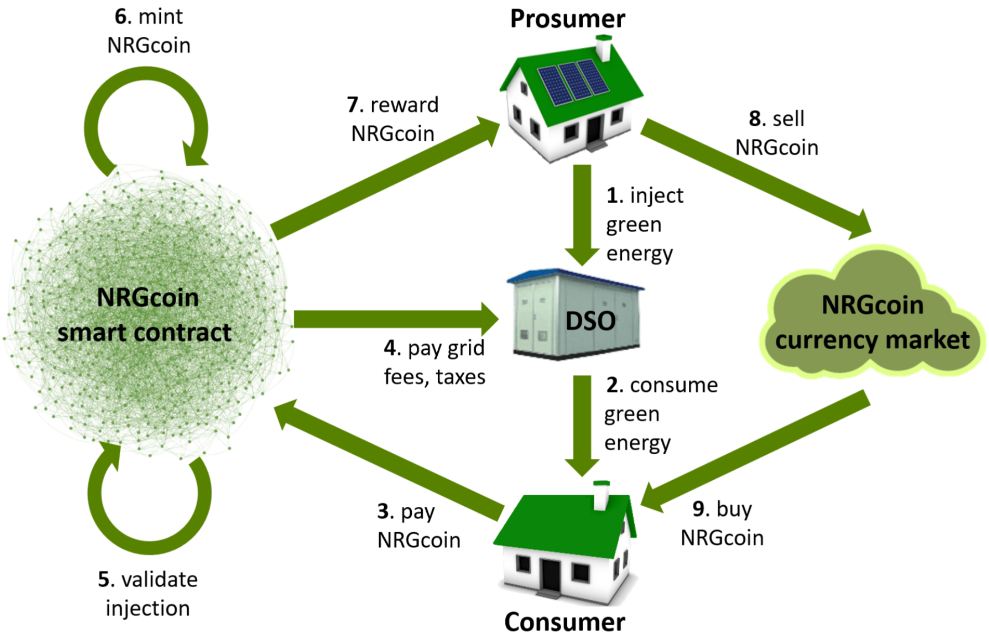


Figure 2.10: NGRCOIN architecture

The NRGcoin project also features a currency market where NRGcoins can be converted into fiat currency and then sold from prosumers or bought by consumers.

Brooklyn Microgrid

Brooklyn Microgrid project was developed by TransactiveGrid which is a joint-venture between Lo3 Energy, a start-up developing blockchain-based applications, and ConsenSys. It was launched in Brooklyn, NY, in 2016, with the main goal of testing the use of blockchain technology for trading energy between neighbours [37].

The main objective of Brooklyn Microgrid testbed was to create a local green energy community where solar panels, installed on the roofs of five residential buildings, supply energy for self-consumption of dwellers, while the surplus is sold to other users in the grid. The system also relies on smart meters to collect data about the amount of energy consumed and produced. All energy trades are managed and recorded via the Ethereum blockchain.

Sunchain

Sunchain solution was created in France and has the objective to certify the energy transactions and trace solar production by tracking and accounting each prosumer for its actual share [38]. Electricity allocation between the participants is automatically executed and certified based on tamperproof conditions using smart contracts stored on the blockchain (figure 2.11) .

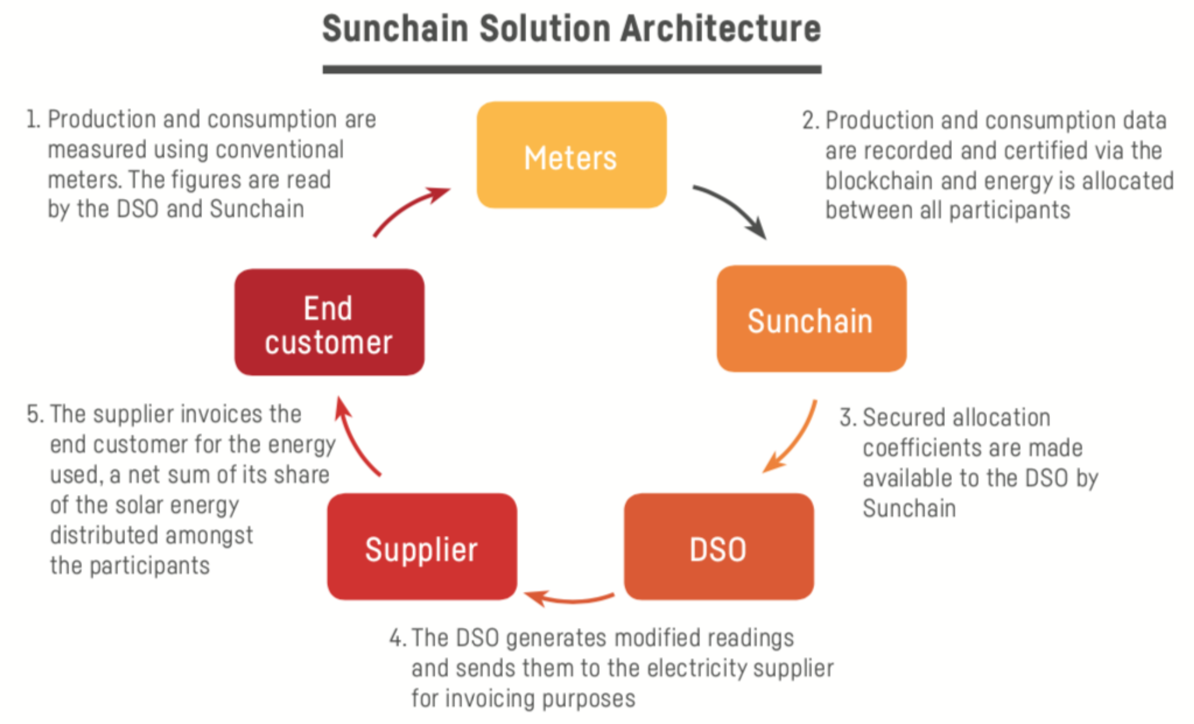

Sunchain Solution Architecture
1. Production and consumption are measured using conventional meters. The figures are read by the DSO and Sunchain
Meters
2. Production and consumption data are recorded and certified via the blockchain and energy is allocated between all participants
End customer
Sunchain
5. The supplier invoices the end customer for the energy used, a net sum of its share of the solar energy distributed amongst the participants
3. Secured allocation coefficients are made available to the DSO by Sunchain
Supplier
DSO
4. The DSO generates modified readings and sends them to the electricity supplier for invoicing purposes

Figure 2.11: Sunchain architecture

SolarCoin

SolarCoin, launched in 2014 by SolarCoin Foundation, is a project aimed at incentivizing the production of renewable energy [39]. It basically consists of a cryptocurrency, SolarCoin, used to reward solar energy producers on the basis of the amount of energy produced. In order to get rewarded for their production, users must prove their contribution by sending production data to the SolarCoin Foundation, which verifies that the claim is valid and that the energy has actually been produced. If so, SolarCoins are transferred to the producer's wallet and he could use them anywhere.

The reward is 1 SolarCoin for every 1 MWh produced from solar and injected into the grid. Origin of every MWh is certified by the blockchain. 98 billion SolarCoins have been created so far, enough to ensure their distribution for 40 years.

Pylon Network

Pylon Network [40] comes from a start-up named KLENERGY TECH, whose solution is being taken into consideration by major energy suppliers like ENDESA. The project proposes to use blockchain technology to help energy sellers have a better knowledge of energy flows. They also developed smart meters that, with blockchain technology, are able to certify energy flows and enable virtual transactions using tokens known as Pyloncoins (figure 2.12). Ethereum blockchain is used also in this project and allows the optimization of energy flows in real-time. The project is based on the Spanish market but the company has already planned to expand internationally.

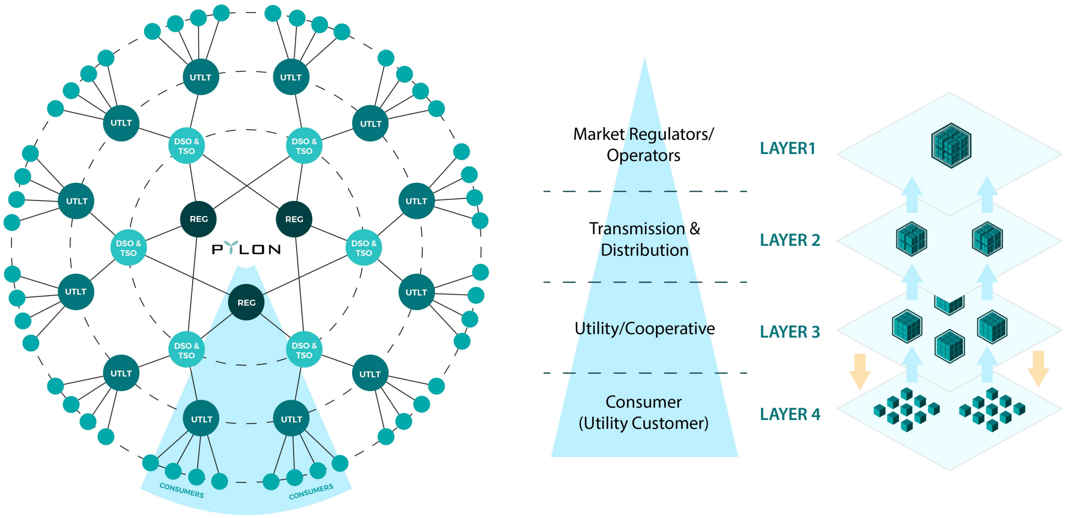


Figure 2.12: Pylon Solution architecture

Grid Singularity

Grid Singularity is a blockchain technology company based in Australia, that is developing an open and decentralized energy data exchange platform, with the supervision of the Energy Web Foundation (EWF). It developed a Decentralised Autonomous Area Agent (D3A) that aims to coordinate small energy prosumers and flexible loads, in an open, trustless and decentralized P2P network. "In this market model, every single device is a direct participant in the electricity market and has the ability to act in its own best interest. The market itself is divided up into numerous local markets, arranged in a hierarchical system. With such a structure in place, trading at local levels is strongly encouraged." [41]. They recently launched the D3A User Interface (UI) platform that allows the user to connect with the D3A exchange engine through a user-friendly app, available at `https://www.d3a.io/`.

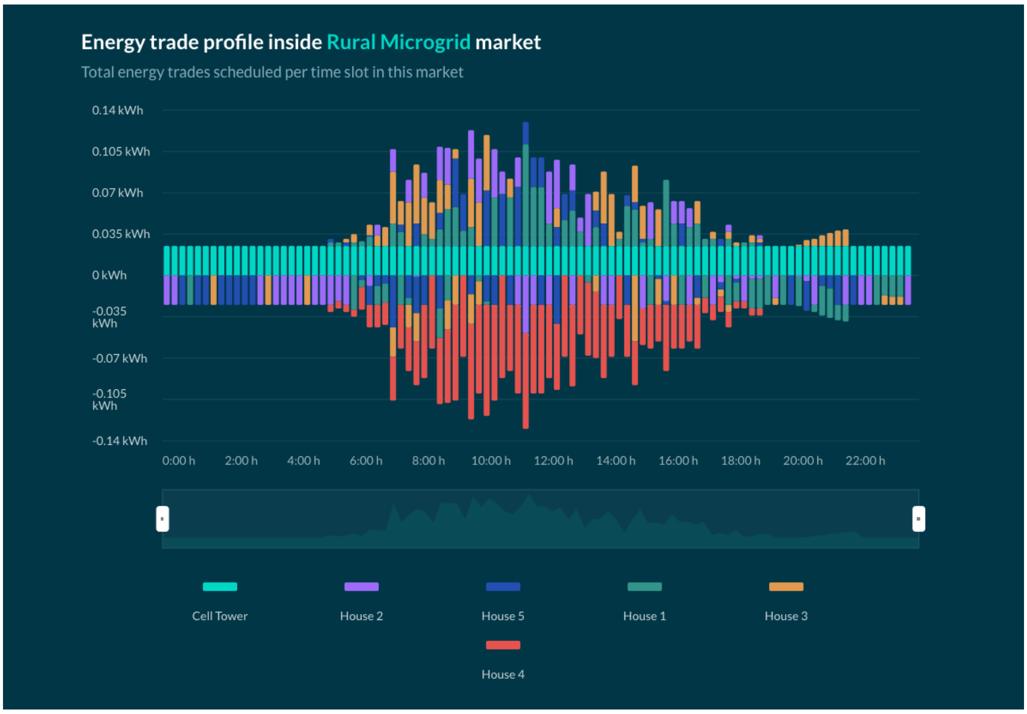


Figure 2.13: D3A UI Interface simulation result on a smart energy grid with five prosumers and a consumer Cell Tower

Energy companies and developers can use the D3A UI to test and showcase the potential of P2P energy trading with homeowners (energy consumers and prosumers) exchanging Renewable Energy Sources (RES), leading to the potential deployment of the D3A exchange engine and consequently fostering the creation of decentralized and distributed smart grid. The figure 2.13 is an example of a simulation of energy exchange that could be made on this platform. It simulates a full 24-hour energy trading between 5 prosumers and a consumer Cell Tower that receives energy from the prosumers and pays them accordingly to the established energy rate. The table 2.1, shows the exact values of energy traded during the simulation.

Table 2.1: Energy bills and net energy traded inside the simulation of the microgrid market

Device / Area	Bought		Sold		Totals	
	Energy (kWh)	Cost (€)	Energy (kWh)	Cost (€)	Energy (kWh)	Cost (€)
House 1	0.756	0.18	0.579	0.18	0.177	0
House 2	0.546	0.13	0.794	0.24	- 0.248	- 0.11
House 3	0.775	0.2	0.402	0.12	0.374	0.08
House 4	0	0	2.45	0.6	- 2.45	- 0.6
House 5	0.462	0.11	0.715	0.21	- 0.253	- 0.1
Cell Tower	2.4	0.73	0	0	2.4	0.73
Accumulated Trades)	4.94	1.34	4.94	1.34	0	0

Grid Singularity also organizes the annual energy blockchain summit, EventHorizon, which is one of the most important events in the field.

3

System design and Implementation

Contents

This chapter presents the steps and decisions made in order to implement the new blockchain solution and create the (Power Share version 2 (PSv2). It is structured as follows:

(a) 3.1 Introduction, where preliminary decisions and first steps made towards the design of the new blockchain system will be detailed; (b) 3.2 Hyperledger Fabric Blockchain Server, where the architecture, settings and business model of the new blockchain system will be described; and (c) 3.3 Power Share Version 2 Android App and ETMS, where the changes made on the server and to previous version of the android app are described.

3.1 Introduction

The main goal of this project is to develop and test a more efficient, blockchain-based backend system for the existing Power Share app and, more in general, improving the overall system performance especially in terms of energy consumption and transaction speed.

The first step was thus identifying the technology to be used instead of IOTA. Two were the options:

- Identify a blockchain-based cryptocurrency to be used instead of mIOTA to make the payments between residents after transactions are performed on the ETMS;
- Use a blockchain technology that enables the use of smart contracts so as to make both transactions and payments between residents automatic and more efficient.

Concerning the first option, the most promising system appeared to be SolarCoin, a cryptocurrency implemented in 2014 with the goal of incentivizing solar PV production. SolarCoin (described in Section 2.3) has become one of the world's largest solar platforms, nonetheless, after some research and testing, we realized it wasn't the most suitable option. Not only because the verification process is complex, but also because it only works with affiliates that have proprietary smart-meter systems and, in our case, replacing the raspberry pi based monitoring system installed in the houses of participants in the project was not possible. Also, the processing time for the confirmation of the solar energy production takes a few days [42] and the system only takes into account the energy produced by prosumers that are then rewarded accordingly, while does not consider the consumers who buy such energy. Another drawback is that smart contracts could not be implemented to regulate energy consumption and production in real-time.

The other solution, implement a blockchain enabled smart contract, would require an extra effort on the development side, having a higher implementation complexity in comparison to the first solution. However, the main advantages of this solution it is that smart contracts could be used to handle transactions and assets with much more efficiency and allow almost real-time payments between participants in the network, thus increasing the overall system performance and stability.

Ethereum (ETH) was the first blockchain allowing to run smart contracts. Nonetheless, further blockchains - with different consensus algorithms and features (see table 3.1) - offering the same opportunity of running smart contracts have been developed and thus were taken into consideration.

Table 3.1: Comparison table between smart contract compatible blockchains

	Ethereum	**Cosmos**	**Cardano**	**EOS**	**Hyperledger**
Token	ETH	ATOM	ADA	EOS	-
Company	The Ethereum Foundation	Interchain Foundation	Cardano Foundation, IOHK, Emurgo	Block.One	Linux
Aim	Become a global decentralized supercomputer	Create the internet of blockchains via interoperability	Create a scientifically backed smart contract platform	Create a scalable platform for industrial scale Dapps	Platform for enterprises to create their own permissioned blockchain
Consensus	POW as of now, will move to Casper POS	Tendermint	Ouroboros	DPOS	PBFT (Practical Byzantine Fault Tolerance)
Smart Contract Code	Solidity (will soon incorporate Vyper)	Language Freedom	Plutus	Web Assembly	Chaincode
ICO	$18.4 million	$17 million	$63 million	$4 billion	-

During the thesis development and after some research and tests, ETH revealed to have the biggest community of experienced developers and is comparatively far developed, with lots of full running DApps, and is well documented. However, it is still using PoW consensus - the planned switch to PoS [27] has not yet happened -, making it not such a viable option due to the high energy costs associated with a network of computers that check and validate the blockchain transactions (PoW consensus algorithm). Also, the transaction cost and speed are further downsides of ETH.

Cosmos and Cardano were other alternatives to take into consideration during the preliminary research. They both have new features and different smart contract coding languages, but, at the same time, are still in a very early phase. Little documentation available and low adoption rate (in comparison with other blockchains) led us to drop both of them from the list of suitable alternatives.

Some tests and advanced research were made on the EOS crypto as well. This system is one of the best in terms of transaction speed and fees are nonexistent. Nonetheless, again, this blockchain platform was launched recently, thus the documentation and support material to write smart contracts is very little. Only at the time of writing of this document the EOSIO v2 has been released, which brings more flexibility, improved security, better performance, and new tools to help developers working with this Blockchain software architecture [43].

Finally, Hyperledger Fabric was the last blockchain solution analyzed as a replacement for the old IOTA-based system. After some tests with Hyperledger Fabric v1.2 and some written javascript chaincode (Hyperledger smart contracts), this option appeared to be the most suitable one. The main factors leading to such choice were (1) the good documentation available, (2) the fact that it is open-source, (3) smart contracts can be written in Java, Javascript and Go. Hyperledger doesn't have its own cryptocurrency, however, the necessary assets (energy tokens, energy value and fiat currency) could be easily tracked and managed on this platform.

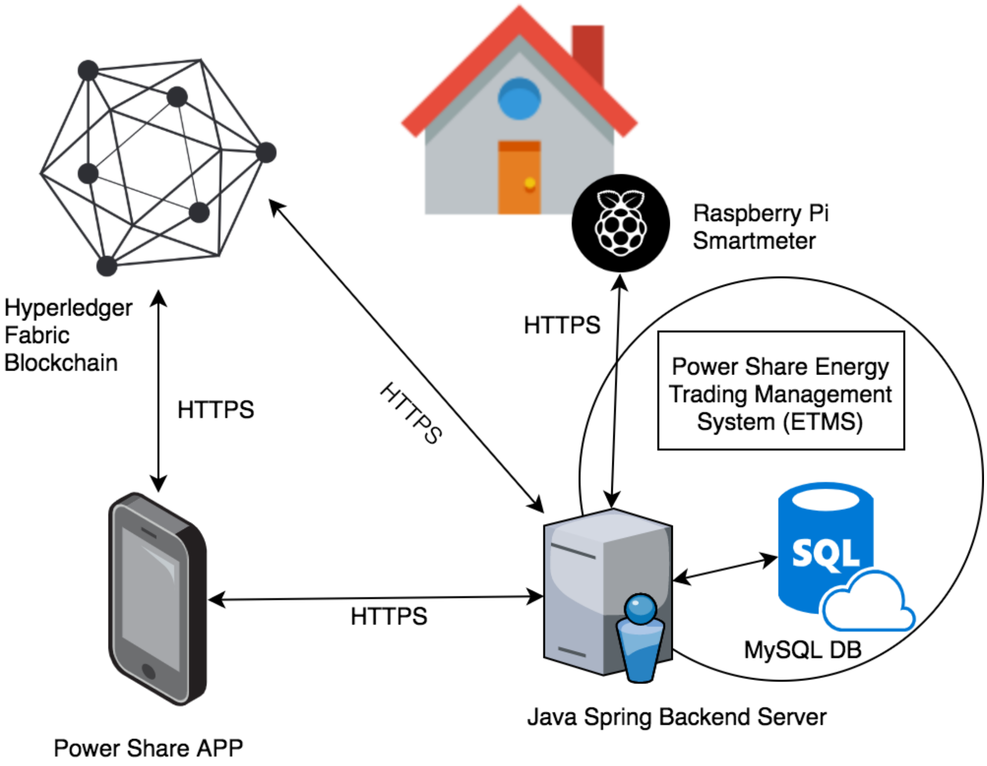


Figure 3.1: PSv2 Backend Architecture

The proposed Backend Architecture (see figure 3.1) consists in the implementation of a blockchain solution based on Hyperledger Fabric 1.2 with development of a secure REST API (using HyperText Transport Protocol Secure (HTTPS)) and additional security methods to communicate with an adapted ETMS java server and PSv2 android app. A new mobile application, adapted in order to work with the new blockchain system replacing the IOTA-based one used in PSv1, will be developed and released.

3.2 Hyperledger Fabric Blockchain Server

3.2.1 Introduction

The Power Share Blockchain Server is a prototype of a decentralized energy network. It is based on the IBM Hyperledger Fabric version 1.2 and created using Hyperledger Composer code pattern available at https://developer.ibm.com/patterns/decentralized-energy-hyperledger-composer/.

The blockchain network allows Residents (small energy prosumers and energy consumers), EVs, and different Utility companies (like the Empresa de Eletricidade da Madeira (EEM)) to exchange energy between each other. Additionally, there is a fourth typology of participants in the network, Banks, which are in charge of exchanging tokens for fiat currencies or vice versa at the current conversion rate.

Three are the assets that can be traded on the blockchain system, (1) Energy, that represents the energy (in kWh) available for trading on the network - i.e. the amount of energy each participant is willing to sell; (2) Coins, that represents the energy tokens owned by participants. Only coins can be used as a mean of exchange for energy; (3) Cash, that represents the amount of fiat currency units owned by participants (like a bank account balance) that can be used to “purchase” coins.

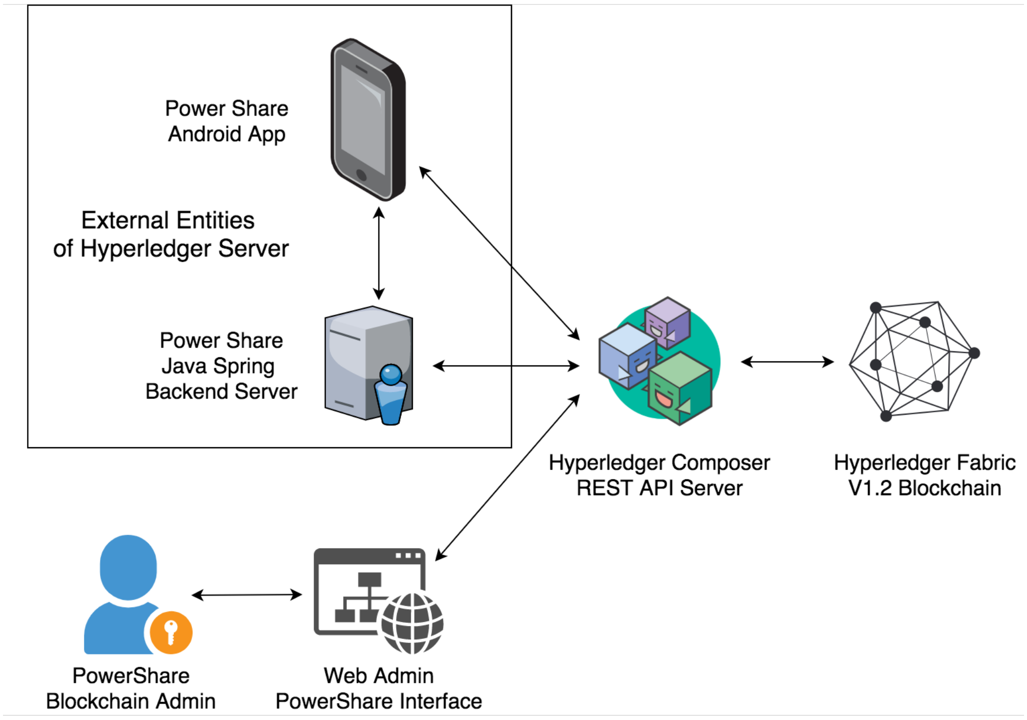


Figure 3.2: PSv2 Hyperledger Composer Rest API Endpoints

The Blockchain System runs on a VPS with Linux Ubuntu 16.04.6 Long Term Support (LTS) server equipped with a 2 core processor and 4GB RAM. The Hyperledger Composer framework was installed on top of the Hyperledger Fabric system "which allows for pluggable Blockchain consensus protocol to ensure that the transactions are validated according to the policy defined by the designated business network participants" [44].

The Blockchain system and Runtime mechanism are implemented on the top of docker containers. Two endpoints (see figure 3.2) are available to interact with the blockchain,(1) a REST API that allows external entities (like the PSv2 android app and the ETMS system) to interact with the blockchain, and (2) the Angular based admin web interface, that is a tool the administrator of the company operating the energy trading network (i.e. Power Share) can use to manage Participants (Residents, EVs, Utility Companies, and Banks), verify assets and transactions within the network, as well as manually perform transactions between Participants.

The REST API for the business network, based on IBM Loopback technology, is generated by the Hyperledger Composer framework (presented at appendix B). Also, the admin web interface Angular application, that also interacts with the REST API, is created using the composer-cli and composer-rest-server plugins of javascript NPM toolset.

3.2.2 Business Model

Hyperledger Composer framework enables developers to easily write a business model definition, using the composer object-oriented modelling language.

The Power Share Hyperledger Composer CTO file (presented at appendix A) contains a single namespace (org.smile.energy.network) and all blockchain resource statements are inherent in this namespace. By using the Composer Modeling Language, the composition of the resources that will be saved in the blockchain - like the blockchain participants, the assets that they could trade and the transactions between them.

3.2.2.A Participants

In the blockchain system we have four different participant types:

- **Resident**: this typology of participants represents the house owners participating in this project (which have smart meters installed in their houses). "Residents" can be prosumers - i.e. users that both produce and consume energy - or simple energy consumers. They are identified on the blockchain system by a residentID and a set of characteristics, namely first name, last name, email, energy buying and selling rates. Residents can own all the three assets (Energy, Coins and Cash) and can trade with fellow residents (buy or sell Energy), EVs (buy or sell Energy, transfer

Coins to/from the EVs), Banks (exchange Cash for Coins and vice versa), and Utility Companies (buy or sell Energy);

- **UtilityCompany**: This participant represents the Utility Companies (energy providers) that are available to sell their energy to Residents or EVs. In this project, the main utility companies are EEM and multiple renewable energy power plants it owns. In the future, the idea is that private investors could assemble a solar Photovoltaic (PV) power plant, a wind farm or another kind of power plant and join the network to sell their energy to other blockchain participants. Utility Companies are identified on the blockchain system by their utilityID and name. They can own two assets (Energy and Coins) and trade with Residents (buy or sell Energy) and EVs (buy or sell Energy);
- **Bank**: This participant is in charge of exchanging tokens for fiat currencies or vice versa at the current conversion rate (which during the project was at 1:1 coin to euro conversion ratio). Banks are identified on the blockchain system by their bankID and name. They own two assets (Cash and Coins) and can trade with Residents (trade Coins for Cash or trade Cash for Coins);
- **Electric Vehicle**: This participant represents EVs, which can trade energy with Residents, Utility companies and even other EVs. The main motivation for adding EVs to the network participants is that they are a subject of interest in the SMILE project, which includes a pilot aimed at the implementation of a low-cost smart charging solution for EVs. They are identified on the blockchain system by an evID and represented by a set of characteristics like electric vehicle type, brand, model, ownerID, energy buying and selling rates. EV are also distinguished between 5 different types,(1) Battery Electric Vehicles (BEV), which are fully electric vehicles with an electric engine and high capacity rechargeable batteries, (2) Plug-in Hybrid Electric Vehicle (PHEV), which have two engines (electric and gasoline) and high capacity rechargeable batteries that could be charged by an outside source, (3) Hybrid Electric Vehicles (HEV), having two engines (electric and gasoline) but a low capacity battery (in comparison to PHEV) that is charged by the car's regenerative braking system, (4) SCOOTER, which represents a new trend in urban mobility, with integrated batteries and designed to be stood upon by the operator, (5) BIKE, which are the most powerful two-wheel vehicles. EVs can own two assets (Energy and Coins) and they can trade with residents (buy or sell Energy), other EVs (buy or sell Energy) and Utility Companies (buy or sell Energy).

3.2.2.B Assets

In our blockchain system we have three different assets:

- **Energy**: This asset represents the energy (the standard measurement unit is in kWh) available for trading within the network. It consists of the amount of energy that each participant is able to sell.

Such asset is identified on the network by an energyID value which is tied to a participant by the ownerID value. Energy asset is used in the EnergyToCoins transactions;

- **Coin**: This asset represents the energy tokens owned by those participants that are allowed to purchase energy on the blockchain network. During this study, the coin value remained tied to the fiat currency (Euros), with a 1:1 exchange rate. This asset is identified on the network by a coinID value and is tied to a participant by its ownerID value. Coin asset is used in TransferCoinsToEV and EnergyToCoins transactions;

- **Cash**: This asset represents the amount of fiat currency units that a participant owns in the blockchain system (the currency used in our study is euros) and can be used to "purchase" coins. The Cash asset is identified on the network by its cashID value and is tied to a participant by the ownerID value. This asset is not restricted to any specific currency, so a user could have euros, dollars or any kind of currency as it's cash asset. Cash is used in the CashToCoins, CashWithdrawalResident and CashFundingResident transactions.

3.2.2.C Transactions

In the blockchain system we have five different typologies of transaction:

- **EnergyToCoins**: This transaction represents the exchange of energy for coins between two blockchain participants. It could be performed by Residents, EVs and Utility Companies. Once the system receives the information regarding energy rate (in Coins/kWh) and energy value (in kWh), it performs the transaction by increasing the coin asset balance of the seller while decreasing his energy balance accordingly. The coin and energy assets of the buyer are updated too;

- **CashToCoins**: This transaction is responsible for the exchange of coins for cash between two blockchain participants. It could be performed by Residents and Banks, where Residents could get or sell coins to the Bank. At this moment, we are using euros as the main currency, but the system has the possibility to be extended to other fiat currencies. Once the system receives data about rate (Coins/EUR) and cash value (in EUR), the transaction is performed i.e. on the buyer side, the coin asset increases while the cash one decreases accordingly. On the seller side, the coin asset balance decreases while the cash balance increases;

- **TransferCoinsToEV**: This transaction is responsible of the coin transfer between a Resident and a EV. Residents could fund or withdraw coins from the coin wallet of an EV. Once the coin value to be transferred is defined, the transaction take place resulting in the decrease of the sender coin balance and a consequent increase of the receiver's coin balance;

- **CashFundingResident**: This transaction connects fiat currencies with the blockchain cash asset. Every time a Resident requests to fund his account, the blockchain administrator will initiate the CashFundingResident transaction, through which the Resident's cash balance is increased accordingly to the amount of fiat currency units provided to the blockchain administrator;

- **CashWithdrawalResident**: This transaction is similar to the CashFundingResident one, but it works the other way around. In this case, the Resident can request the blockchain administrator to withdraw money from his account and receive the corresponding amount of fiat currency in return while the system updates his coin balance accordingly.

3.2.3 Business Network

After the business model is defined, the Hyperledger Composer framework allows developers to build a complete business network (figure 3.3) and write the chaincode (i.e. Hyperledger Fabric smart contracts).

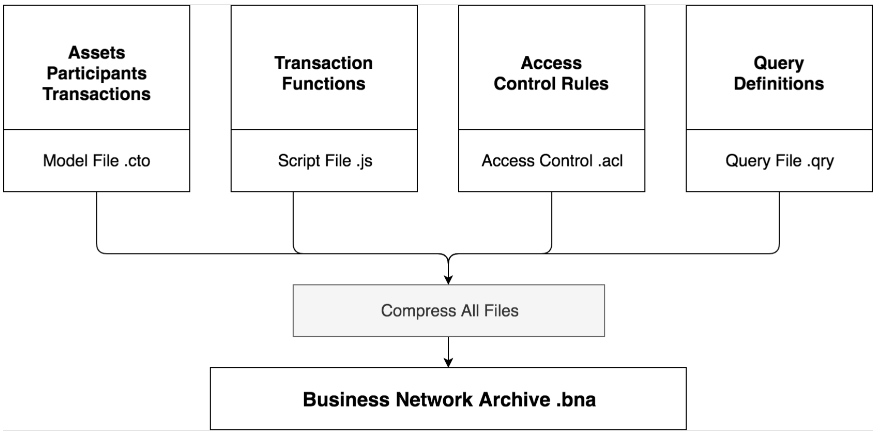


Figure 3.3: Hyperledger Composer business network representation

A Business network archive file (in our case it corresponds to the smile-energy-network@0.1.16.bna file), consisting of a package of the following four kinds of files, is created:

- **Model file**, described in section 3.2.2, which contains all the defined blockchain transactions, assets and participants (appendix A);

- **Script File** (transactions.js), which contains all the JavaScript code of the Hyperledger chaincode describing the transactions logic in the blockchain. Some few unit tests were made to ensure that the smart contracts work properly;

- **Access File** (permissions.acl), containing all the blockchain participants permission rules. Access Control Language (ACL) is used to express permissions rules where, in short, all participants in the network have read access rights to all assets and other participants values but update and delete right is guaranteed only to their own account. Only the blockchain system admin has all right (read, update and delete) overall assets and participants;
- **Query File** (queries.qry), which contains all the blockchain database queries that allow to show all transactions performed, participants and assets on the blockchain.

3.2.4 Power Share Admin Interface

The Blockchain Admin Interface (figure 3.4, called Smile Hyperledger Interface, is a tool for the administrator of the company operating the energy trading network – i.e. Power Share. It can be used to manage Participants (Residents, EVs, Utility Companies, and Banks), verify assets and transactions within the network, as well as manually perform transactions between Participants. Residents could also request the administrator to fund or withdraw funds (cash) from their account. The Smile Hyperledger Interface is based on Angular.JS and bootstrap framework for CSS styling.

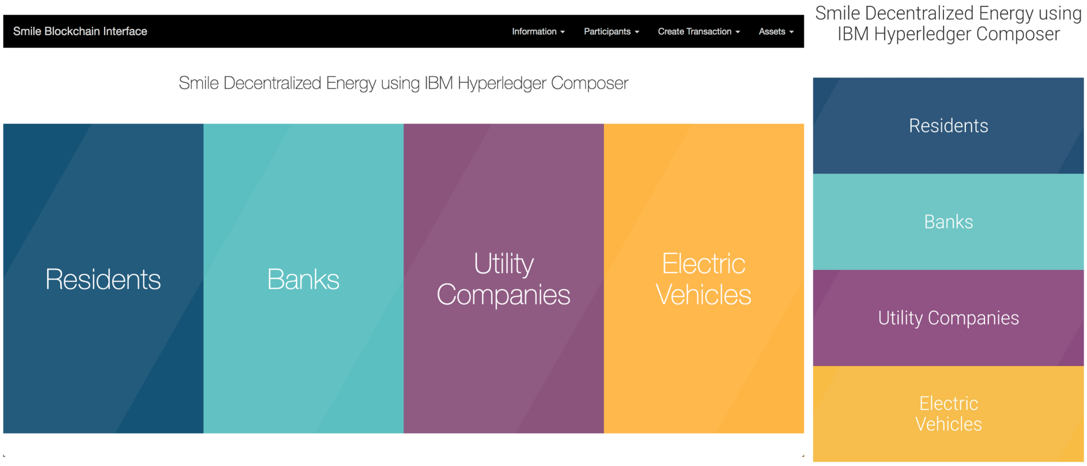
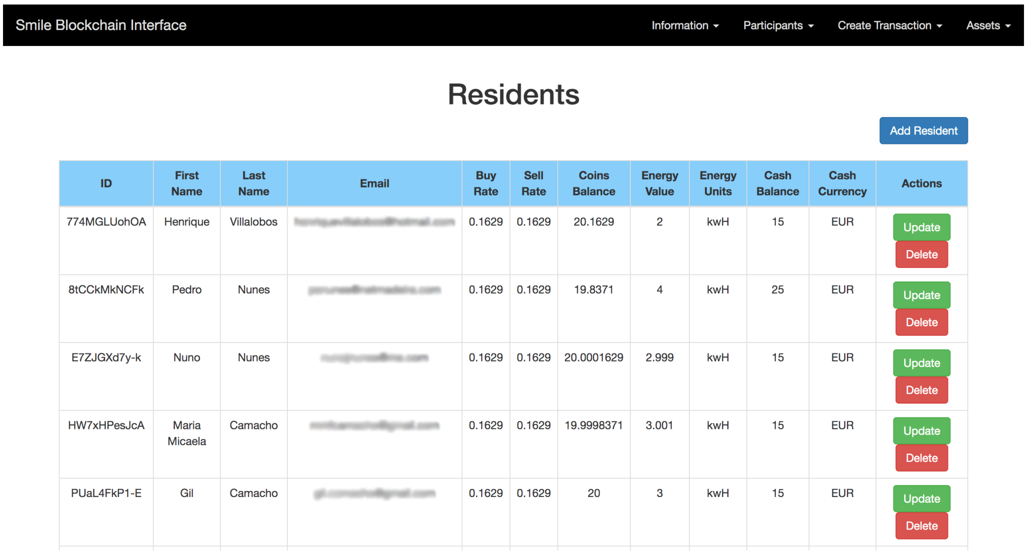


Figure 3.4: Admin interface home screen for laptop and smartphone screens

Manage participants data

The admin interface has a feature that shows data of each blockchain participants (figure 3.5). It's divided into four sections, each one corresponding to a participant type, and allows the blockchain admin to check and update all participant data and respective assets. It also allows to add a new participant or delete an existing one.



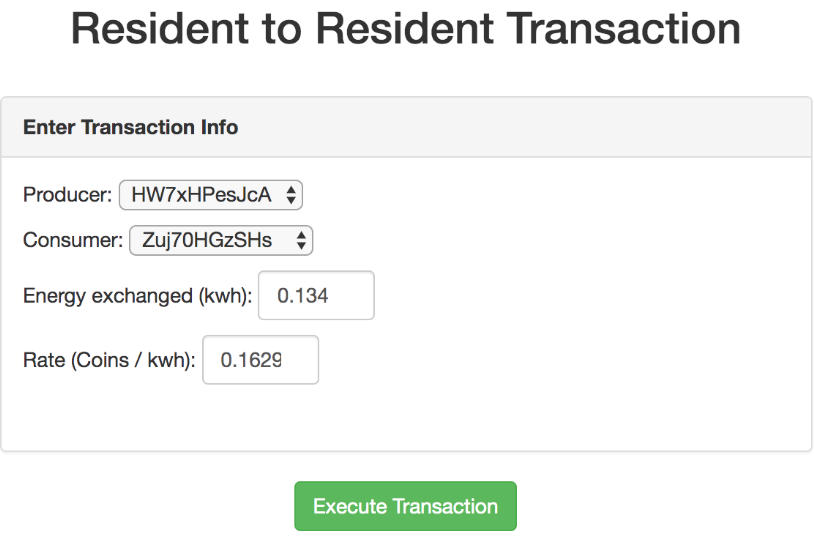
Residents

ID	First Name	Last Name	Email	Buy Rate	Sell Rate	Coins Balance	Energy Value	Energy Units	Cash Balance	Cash Currency	Actions
774MGLUohOA	Henrique	Villalobos	[redacted]	0.1629	0.1629	20.1629	2	kwH	15	EUR	Update Delete
8tCCkMkNCFk	Pedro	Nunes	[redacted]	0.1629	0.1629	19.8371	4	kwH	25	EUR	Update Delete
E7ZJGXd7y-k	Nuno	Nunes	[redacted]	0.1629	0.1629	20.0001629	2.999	kwH	15	EUR	Update Delete
HW7xHPesJcA	Maria Micaela	Camacho	[redacted]	0.1629	0.1629	19.9998371	3.001	kwH	15	EUR	Update Delete
PUaL4FkP1-E	Gil	Camacho	[redacted]	0.1629	0.1629	20	3	kwH	15	EUR	Update Delete

Figure 3.5: Residents information screen

Create new transactions on the blockchain

The next section of the main menu opens the create transaction feature, that allows the blockchain administrator to perform manual transactions on the blockchain. An example of a resident to resident energy transaction is represented by figure 3.6. In this case, it opens an input field that allows to define seller, buyer, amount of energy exchanged (in kWh) and transaction rate (Coins / kWh). Obviously, the parameters in the input field are slightly different for each typology of transaction.



Resident to Resident Transaction

Enter Transaction Info

Producer: HW7xHPesJcA

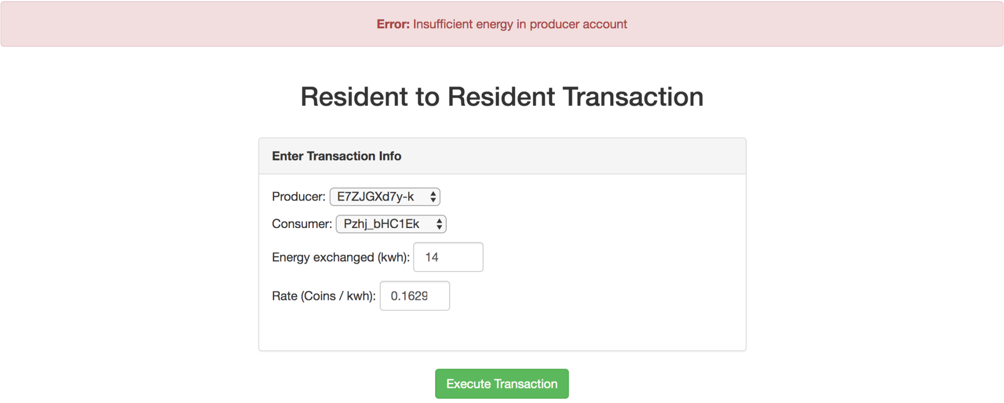
Consumer: Zuj70HGzSHs

Energy exchanged (kwh): 0.134

Rate (Coins / kwh): 0.1629

Figure 3.6: Resident to Resident transaction input

If anything goes wrong, the system will display an error message below the menu bar. In the example presented in the figure 3.7 (a resident to resident energy transfer), the system verified beforehand the amount of energy stored in the battery of the seller and, since it was not enough to meet the buyer demand, it displayed the error message ”Insufficient energy in producer account” and the transaction did not proceed. In the same way, if the buyer doesn’t have enough coins to buy the amount of energy required, the system displays the ”Insufficient coins in consumer account” error message. Also, if a server error or connection error appears, a ”Could not connect to REST server” message will appear. If the blockchain interface could not connect to the Hyperledger API, it will show the error message ”404 - Could not find API route.”



Error: Insufficient energy in producer account

Resident to Resident Transaction

Enter Transaction Info

Producer: E7ZJGXd7y-k

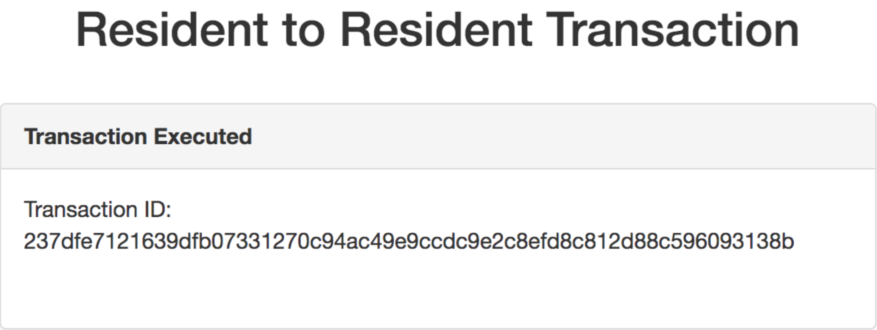
Consumer: Pzhj_bHC1Ek

Energy exchanged (kwh): 14

Rate (Coins / kwh): 0.1629

Figure 3.7: Resident to Resident transaction screen with error “Insufficient energy in producer account”

After passing these checks, the API request is sent to the Hyperledger server, which will return the string ID of the transaction made on the blockchain (figure 3.8).



Resident to Resident Transaction

Transaction Executed

Transaction ID:
237dfe7121639dfb07331270c94ac49e9ccdc9e2c8efd8c812d88c596093138b

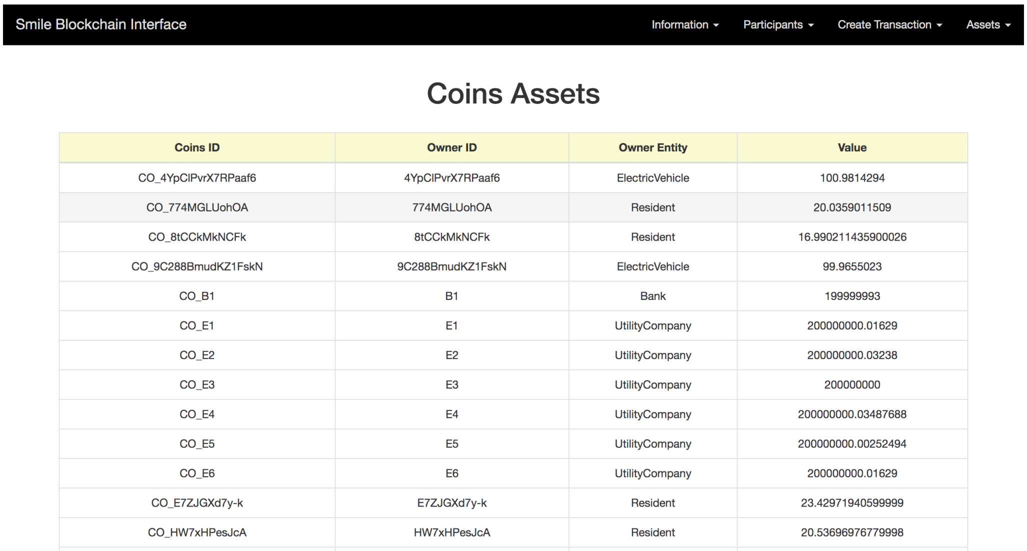
Figure 3.8: Resident to Resident successful transaction confirmation with transaction ID

All the transaction requests on the admin panel menu are related to the five different transaction types available in the blockchain:

- The **EnergyToCoins** transaction includes Resident to Resident, Resident to Utility Company, Resident to Electric Vehicle, Electric Vehicle to Utility Company and Electric Vehicle to Electric Vehicle;
- The **CashToCoins** transaction corresponds to Resident to Bank;
- The **TransferCoinsToEV** transaction consists in the Resident to Electric Vehicle Coin Transfer;
- The **CashFundingResident** transaction represents the Resident Cash Funding;
- On **CashWithdrawalResident** transaction represents the Resident Cash Withdraw.

Check assets

The menu item called "assets" allows the blockchain administrator to check all assets owned by all participants on the blockchain (figure 3.9). This is a read-only feature that allows the admin to check which asset id belongs to which participant, as well as its value/balance and characteristics (e.g. kWh for the energy asset and EUR for the cash asset).



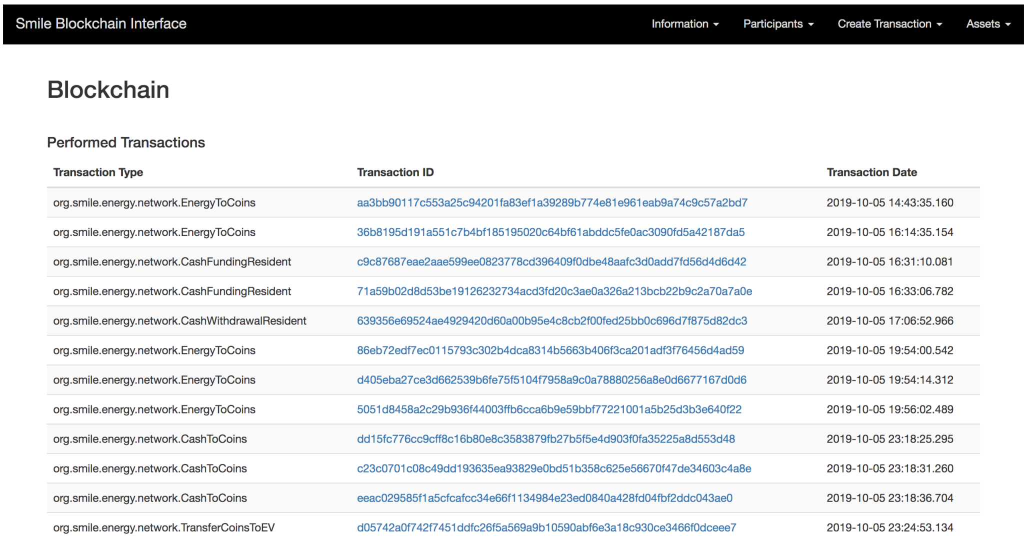
Coins Assets

Coins ID	Owner ID	Owner Entity	Value
CO_4YpClPvrX7RPaaf6	4YpClPvrX7RPaaf6	ElectricVehicle	100.9814294
CO_774MGLUohOA	774MGLUohOA	Resident	20.0359011509
CO_8tCCkMkNCFk	8tCCkMkNCFk	Resident	16.990211435900026
CO_9C288BmudKZ1FskN	9C288BmudKZ1FskN	ElectricVehicle	99.9655023
CO_B1	B1	Bank	199999993
CO_E1	E1	UtilityCompany	200000000.01629
CO_E2	E2	UtilityCompany	200000000.03238
CO_E3	E3	UtilityCompany	200000000
CO_E4	E4	UtilityCompany	200000000.03487688
CO_E5	E5	UtilityCompany	200000000.00252494
CO_E6	E6	UtilityCompany	200000000.01629
CO_E7ZJGXd7y-k	E7ZJGXd7y-k	Resident	23.42971940599999
CO_HW7xHPesJcA	HW7xHPesJcA	Resident	20.53696976779998

Figure 3.9: Coin asset screen where all the coin assets in the blockchain are displayed

Check all transactions performed on the blockchain

The menu item “Information” opens a drop-down menu containing two sections, one is the about page with some details explaining how the network works, and the other “Blockchain transactions” allows the user to go and check a list of all transactions performed on the blockchain. The transactions are ordered by timestamp. For each, information about transaction type, IDs and date is displayed as per (figure 3.10).



Blockchain

Performed Transactions

Transaction Type	Transaction ID	Transaction Date
org.smile.energy.network.EnergyToCoins	aa3bb90117c553a25c94201fa83ef1a39289b774e81e961eab9a74c9c57a2bd7	2019-10-05 14:43:35.160
org.smile.energy.network.EnergyToCoins	36b8195d191a551c7b4bf185195020c64bf61abddc5fe0ac3090fd5a42187da5	2019-10-05 16:14:35.154
org.smile.energy.network.CashFundingResident	c9c87687eae2aae599ee0823778cd396409f0dbe48aafc3d0add7fd56d4d6d42	2019-10-05 16:31:10.081
org.smile.energy.network.CashFundingResident	71a59b02d8d53be19126232734acd3fd20c3ae0a326a213bcb22b9c2a70a7a0e	2019-10-05 16:33:06.782
org.smile.energy.network.CashWithdrawalResident	639356e69524ae4929420d60a00b95e4c8cb2f00fed25bb0c696d7f875d82dc3	2019-10-05 17:06:52.966
org.smile.energy.network.EnergyToCoins	86eb72edf7ec0115793c302b4dca8314b5663b406f3ca201adf3f76456d4ad59	2019-10-05 19:54:00.542
org.smile.energy.network.EnergyToCoins	d405eba27ce3d662539b6fe75f5104f7958a9c0a78880256a8e0d6677167d0d6	2019-10-05 19:54:14.312
org.smile.energy.network.EnergyToCoins	5051d8458a2c29b936f44003ffb6cca6b9e59bbf77221001a5b25d3b3e640f22	2019-10-05 19:56:02.489
org.smile.energy.network.CashToCoins	dd15fc776cc9cff8c16b80e8c3583879fb27b5f5e4d903f0fa35225a8d553d48	2019-10-05 23:18:25.295
org.smile.energy.network.CashToCoins	c23c0701c08c49dd193635ea93829e0bd51b358c625e56670f47de34603c4a8e	2019-10-05 23:18:31.260
org.smile.energy.network.CashToCoins	eeac029585f1a5cfcafcc34e66f1134984e23ed0840a428fd04fbf2ddc043ae0	2019-10-05 23:18:36.704
org.smile.energy.network.TransferCoinsToEV	d05742a0f742f7451ddfc26f5a569a9b10590abf6e3a18c930ce3466f0dceee7	2019-10-05 23:24:53.134

Figure 3.10: Hyperledger blockchain transactions list

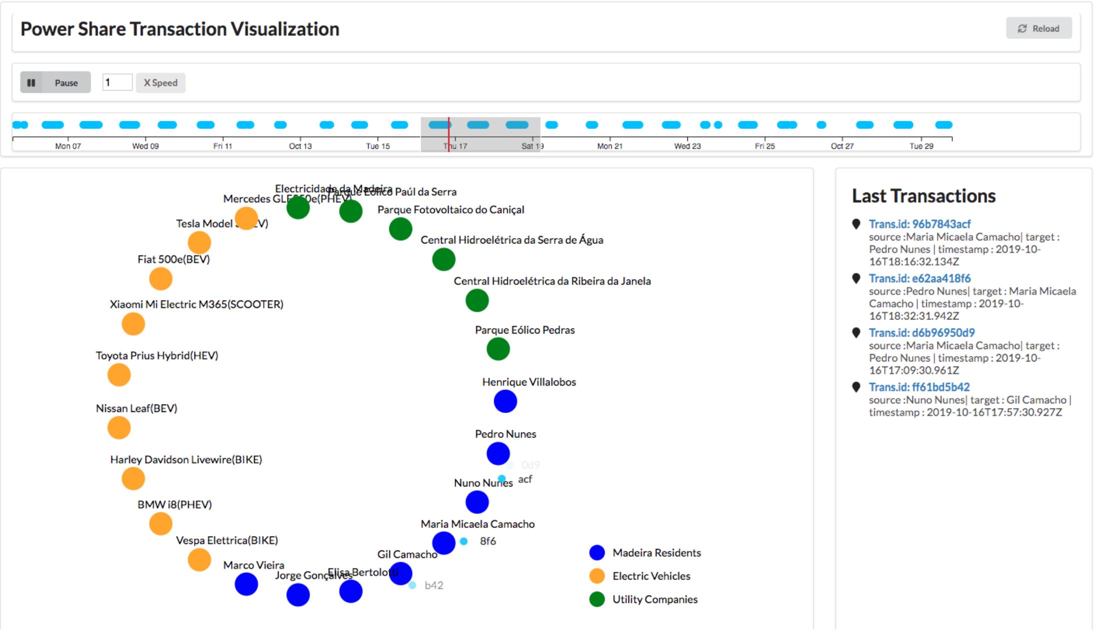
The transaction ID is clickable and leads to access further information which is requested to the Hyperledger REST API (listing 3.1).

Listing 3.1: JSON REST API response with detailed information of one EnergyToCoins transaction

```
{
    ”$class”:”org.smile.energy.network.EnergyToCoins”,
    ”energyRate”:0.1629,
    ”energyValue”:0.020789666666666668,
    ”coinsInc”:”resource:org.smile.energy.network.Coins#CO_HW7xHPesJcA”,
    ”coinsDec”:”resource:org.smile.energy.network.Coins#CO_8tCCkMkNCFk”,
    ”energyInc”:”resource:org.smile.energy.network.Energy#EN_8tCCkMkNCFk”,
    ”energyDec”:”resource:org.smile.energy.network.Energy#EN_HW7xHPesJcA”,
    ”transactionId”:”8f40807bc45324d497e1d58c4d2e106da26babd52cce2f2a4a155c1878979514”,
    ”timestamp”:”2019-10-21T09:12:32.468Z”
}
```

Also, the Hyperledger system transactions (actions like add, delete or update participants and/or assets), which are not described on the network model transactions, are recorded on the blockchain and displayed in this page, right below the above mentioned list.

In order to represent the dynamics behind the energy exchanged - like participants involved, number and frequency of transactions, etc. - a dynamic and interactive visualization was created. It consists of a javascript D3.js page which displays transactions data by making API calls to the Hyperledger REST API server. EnergytoCoins transactions are provided by the server in the JSON format and organized by its timestamp (figure 3.11)



Last Transactions

- Trans.id: 96b7843acf
  source :Maria Micaela Camacho| target : Pedro Nunes | timestamp : 2019-10-16T18:16:32.134Z
- Trans.id: e62aa418f6
  source :Pedro Nunes| target : Maria Micaela Camacho | timestamp : 2019-10-16T18:32:31.942Z
- Trans.id: d6b96950d9
  source :Maria Micaela Camacho| target : Pedro Nunes | timestamp : 2019-10-16T17:09:30.961Z
- Trans.id: ff61bd5b42
  source :Nuno Nunes| target : Gil Camacho | timestamp : 2019-10-16T17:57:30.927Z

Figure 3.11: EnergytoCoins transaction visualization

3.2.5 Server Configurations

3.2.5.A Hyperledger Fabric

In the blockchain server configuration, some steps were taken in order to adapt it to the smile project namespace. After setting the Hyperledger fabric version to 1.2, in the composer folder (at /fabric-dev-servers/fabric-scripts/hlfv12/composer) two configuration files were set in order to define and create the Hyperledger network organization, the network components, as well as users and communication channel.

The cryptogen and configtxgen Hyperledger Fabric configuration tools will read these files and create the digital certificate infrastructure for our blockchain network (the cryptogen tool does that by reading its crypto-config.yaml configuration file), which is defined here as smile.com network. All the admin and blockchain keys generated and digital certificates will be saved on the crypto-config folder (at /fabric-dev-servers/fabric-scripts/hlfv12/composer/crypto-config).On the crypto-config.yaml file, the Hyperledger blockchain orderer and organization were defined with the smile.com and org1.smile.com domains respectively. Only one admin (admin@org1.smile.com) was defined in the Hyperledger Fabric blockchain structure. Since this is a test environment, we use the cryptogen tool to self generate all the security certificates and keys, however, in a more strict and controlled environment (production network), all the certificates and keys should be inserted manually or obtained from external certificate authorities.

The second tool, the configtxgen, is launched after the cryptogen tool has set up all the necessary security keys and certificates by reading from its configtx.yaml configuration file. The blockchain orderer service definition (running with the address orderer.smile.com at the port 7050), the anchor peer namespace (peer0.org1.smile.com) and port (7051), and the communication channel configurations are defined during this process where configuration blockchain artefacts are created.

After running Hyperledger Fabric configuration tools, Hyperledger Composer reads all the configuration from the DevServer_connection.json file. These definitions set up all the network components that connect to the Fabric blockchain network.

The last process consists in generating the business network card for the Hyperledger Fabric administrator (admin@org1.smile.com) and importing the card into the system, thus allowing the administrator to deploy the business network (defined in the section 3.2.3) to the blockchain peer node.

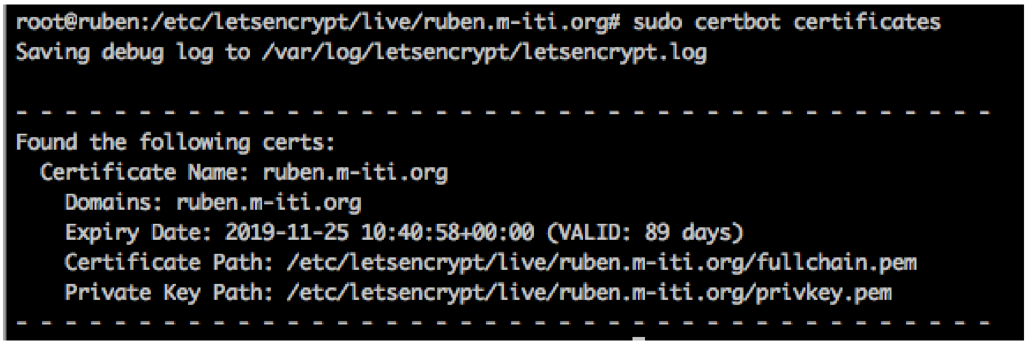
3.2.5.B Server Security

Regarding the overall server security, the Hyperledger blockchain is protected with all the certificates and admin keys, where only the admin of the system has all the permissions to work with the blockchain. However, the REST API endpoint is a vulnerable point in our system, so some primary security measurements were taken.

As a first layer of security an API key was added to the REST API server. By doing this, the server only accepts request connections that include an x-api-key Header with the value of the defined API key and rejects all other request connections. This way, we know that only devices that know the API key can interact with the blockchain.

The HTTPS protocol was used as a second layer of security in order to prevent classical network attacks, like the man-in-the-middle attack, where an attacker could see all the request's information including the x-api-key Header with the API key and transaction data or request.

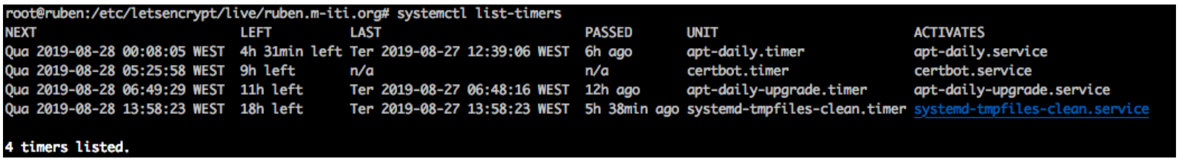
The implementation of the Secure Sockets Layer (SSL) certificates was done by installing Let's Encrypt and Certbot services (figure 3.12). "Let's Encrypt and Electronic Frontier Foundation's Certbot aim to improve the Transport Layer Security (TLS) ecosystem by offering free trusted certificates (Let's Encrypt) and by providing user-friendly support to configure and harden TLS (Certbot)" [45].



```
root@ruben:/etc/letsencrypt/live/ruben.m-iti.org# sudo certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Found the following certs:
  Certificate Name: ruben.m-iti.org
    Domains: ruben.m-iti.org
    Expiry Date: 2019-11-25 10:40:58+00:00 (VALID: 89 days)
    Certificate Path: /etc/letsencrypt/live/ruben.m-iti.org/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/ruben.m-iti.org/privkey.pem
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 3.12: Let's Encrypt Certificate and Private key paths

This allows the linux server to generate new certificates when they are first created, then a cron job is added into the system where it runs a script (figure 3.13) that will renew the certificates every 3 months automatically before they expire (each Let's Encrypt certificate has a 3 months validity).



```
root@ruben:/etc/letsencrypt/live/ruben.m-iti.org# systemctl list-timers
NEXT                          LEFT          LAST                          PASSED       UNIT                         ACTIVATES
Qua 2019-08-28 00:08:05 WEST  4h 31min left Ter 2019-08-27 12:39:06 WEST  6h ago       apt-daily.timer              apt-daily.service
Qua 2019-08-28 05:25:58 WEST  9h left       n/a                           n/a          certbot.timer                certbot.service
Qua 2019-08-28 06:49:29 WEST  11h left      Ter 2019-08-27 06:48:16 WEST  12h ago      apt-daily-upgrade.timer      apt-daily-upgrade.service
Qua 2019-08-28 13:58:23 WEST  18h left      Ter 2019-08-27 13:58:23 WEST  5h 38min ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service

4 timers listed.
```

Figure 3.13: Certbot.service systemctl cronjob

3.3 Power Share Version 2 Android App and ETMS

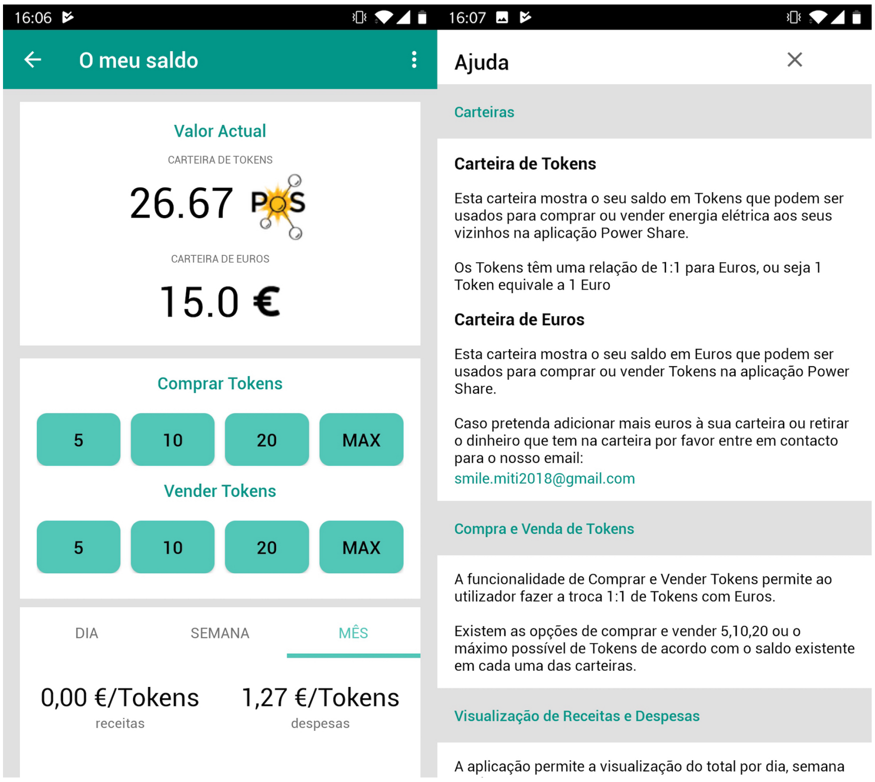
Before releasing the new version of the Power Share system, changes on both the ETMS java server and android app were made.

As a first step, all the ETMS server problems and the reasons why it was not receiving any data from the smart meters, were identified. After some tests, it emerged that the database table was full. Also, we noticed that the "HTTP 200 OK" response to smart meters (which allows them to know that the energy data was successfully received on the ETMS) was taking too long to be generated. Sometimes it surpassed the timeout time defined on the smart meters, leading them to think that the server was offline and thus save the energy data into their own database. After each minute the JSON file is updated with all the energy readings that were not saved into the ETMS system, resulting in an even worse scenario for recovery where no energy data is stored, because the JSON sent to the ETMS is then huge. Besides, some smart meters were offline, so it was necessary to run a check on every device to know its status and identify the problem (if any).

After checking all the available smart meters, cleaning some of the database tables, improving server code and replicating the database to a new one for this experiment, data sent from the smart meters started to be saved as they should without errors.

Some tweaks to the ETMS system were made to remove the IOTA technology that was implemented on the PSv1 and replace it so as to connect the system to the new Hyperledger Blockchain Server. Instead of using IOTA addresses and generating new energy transactions associated with the IOTA accounts, transactions are sent to the Hyperledger Blockchain API and payments and assets are updated automatically.

GNU Screen application was also installed on both ETMS and blockchain servers, allowing to create virtual terminals inside the Secure Shell (SSH) connection, where the application processes will be initiated. After terminating the SSH session, it's possible to resume the screen session and check all the application process information as if the session was always open.



Valor Actual

CARTEIRA DE TOKENS

26.67

CARTEIRA DE EUROS

15.0 €

0,00 €/Tokens
receitas

1,27 €/Tokens
despesas

Carteiras

Carteira de Tokens

Esta carteira mostra o seu saldo em Tokens que podem ser usados para comprar ou vender energia elétrica aos seus vizinhos na aplicação Power Share.

Os Tokens têm uma relação de 1:1 para Euros, ou seja 1 Token equivale a 1 Euro

Carteira de Euros

Esta carteira mostra o seu saldo em Euros que podem ser usados para comprar ou vender Tokens na aplicação Power Share.

Caso pretenda adicionar mais euros à sua carteira ou retirar o dinheiro que tem na carteira por favor entre em contacto para o nosso email:
smile.miti2018@gmail.com

Compra e Venda de Tokens

A funcionalidade de Comprar e Vender Tokens permite ao utilizador fazer a troca 1:1 de Tokens com Euros.

Existem as opções de comprar e vender 5,10,20 ou o máximo possível de Tokens de acordo com o saldo existente em cada uma das carteiras.

Visualização de Receitas e Despesas

A aplicação permite a visualização do total por dia, semana

Figure 3.14: PSv2 "Ver o meu saldo" screen and help subsection

After working on the ETMS system and guaranteeing its stability, the PSv2 android app was developed. The IOTA technology and its wallet were removed and the “ver o meu saldo” feature was replaced by the Hyperledger blockchain wallet (figure 3.14).

The screen shows two virtual wallets that correspond to the user’s assets on the Hyperledger blockchain server, one for Coins (here represented as energy tokens which are used as a mean of exchange for energy) and the other for Cash (EUR).

Unlike the previous technology, no passwords or seed inputs are needed, since all that the user has to do is buying or selling energy tokens. On this screen, users can also check their daily, weekly and monthly transaction incomes and outcomes. If a user wants to withdraw or fund their account, a request to the blockchain admin needs to be made.

Also, payments are fully automated with this new blockchain system and there is no need to generate payment addresses or manually request to make payments (in the previous version, indeed, users had to click on the “pagar as minhas despesas” button in order to make payments).

The overall app performance and stability was improved. User images on the server were compressed and load faster on the app. The only reported bug that originated a crash during the test on the PSv1 - i.e. a user opened the app in the timeframe between midnight and 2 am. Since the application was requesting the server API for data with wrong intervals, the server returned an error where the application crashed [5] - was solved.

The same Model View - View Model (MVVM) design pattern was used for the development of PSv2 and is fully responsive to all smartphone screens and tablets. In order to update the new version of the Power Share app, the apk was signed (Jar Signature) with the key present at the java Keystore file (smileandroid.jks) which allowed the PSv1 app users to upgrade to the new version (with the same package name) without any need to delete data or make a new login.

4

Evaluation

Contents

This chapter describes the methodology adopted to assess the different components of this project, which have been tested with real users: the PSv2 android app, ETMS and Hyperledger blockchain server. Data collected throughout the study and results from the deployment are presented and briefly discussed.

4.1 Evaluation procedure and testing

Different methodologies have been used to test the system, adopting both quantitative and qualitative approach.

Three are the components of the system that have been tested:

- **The Hyperledger SMILE Admin Interface:** in order to assess the UI usability and collect some general feedback to improve it, we opted for adopting a qualitative approach - i.e. we performed a usability test (consisting of six tasks) using the think-aloud method;
- **The end user PSv2 Android application:** despite it was initially planned, we didn't conduct any usability test with the project participants. This choice is due to the fact that changes on the UI and navigation flow were very minimal. The new version of the app was merely subjected to a preliminary usability assessment performed by an interaction design expert (not reported here), and then evaluated with a quantitative assessment. Specifically, in order to assess application performance and usage patterns, we electronically monitored the system and the interactions with it throughout the study by using the Fabric.io framework;
- **The system backend:** a quantitative approach was used to evaluate the system backend as well - i.e. data concerning its performance and energy consumption were collected and compared to those collected during the deployment of PSv1.

4.2 Qualitative assessment: the Blockchain Admin Interface

4.2.1 Methodology

In order to assess usability and collect feedback on the Administrator Interface a small user test involving 5 participants (males, age ranging from 26 to 35 years, with a background in computer science and informatics) recruited among researchers at the Interactive Technologies Institute of Madeira was conducted. The test aimed mainly at assessing perceived usability, ease of use and clarity of the system, and was structured in two main blocks. First, participants have been asked to perform the following tasks using the web-platform:

1. Read the platform description and comment on it;
2. Check and describe (typology, timestamp, participants and assets involved) of the last transaction performed;
3. Check the status (coins, cash and energy balance) of a given Resident;
4. Add a participant to the network;
5. Perform a manual transaction (transfer energy from an EV to a utility company);
6. Perform a manual transaction (make a resident withdraws cash).

Feedback was collected using the think-aloud method. Finally, a few general questions regarding positive/negative aspects of the system, perceived usability and ease of use, and structure of the platform have been asked. With explicit permission of the participants, the test was audio-recorded. For the purpose of this test, i.e. in order not to affect the system deployment, a new remote VPS Linux server was created. The test server is based in London, it has similar characteristics as the production server and is loaded with all the Hyperledger blockchain applications.

4.2.2 Results

All participants in the user test of the Blockchain Admin Interface had no problems performing tasks 1 (read the platform description), 3 (check participant status) and 5 (manual transaction: sell energy). Two participants couldn't complete task 4 (add a participant) and one failed in performing task 6 (withdraw cash), while four out of five encountered some minor difficulties with task 2 (check the last transaction in the network) but only due the menu items labelling. The failure in task 4 was a system failure – i.e. the participant did perform the task correctly.

Task 1 - read the platform description and comment on it: as stated above, all participants in the study managed to find the information and reported to be able to understand how the system works. Only minor suggestions to improve the UI have been collected, the most relevant one consists in adding some diagrams or visual representations showing how the different participants and assets interact/connect with each other - "*This participants part, ideally would be described through a diagram, where you have arrows and so on. Because it is really hard to retain who can trade with whom. I need to make a drawing myself*" (Participant 1).

Task 2 - check and describe the last transaction performed: main usability issues were found while performing this task. The main bottleneck relates to the menu item it belongs to. The list of transactions,

together with the system description, can be found under the "information" section. To make it easier for the user to reach this page, participants suggested to move it under a more general "transactions" section (which should also include "create transactions") or keep it as a separate item of the top bar menu. After finding the page, a further despite minor issue emerged. Indeed, it took participants some time to find the last transaction among those listed in the page. This because it is displayed on the bottom of the list and not on the top: "*There is no filtering tool here and it is like reverse order. Usually, when you check your bank account for instance, the last thing is in first place*" (Participant 1).
Also, users reported to find difficult finding all information describing the transaction "*I should know all the class items and their key order to retrieve the information, but without knowing that I don't understand. I was expecting some kind of table thing, not this class thing*" (Participant 4).

Task 3 - check the status of a Resident: all participants completed the task. Only minor suggestions emerged. The most relevant ones relate to the need of a sorting tool (e.g. to sort Residents on the basis of their energy balance) and rewording some of the labels in the table: "*Energy Value' is not clear. I'd use another word*" (Participant 3).

Task 4 - add a participant to the network: 3 out of 5 participants performed this task smoothly. The only minor issue reported was the need of refreshing the page in order to see the new network member in the list: "*You should not need to refresh the page*" (Participant 3). As mentioned before, two out of five couldn't add a participant to the network. Both of them performed the task correctly. The issue was indeed on the server-side. Such server error could be due to the characteristics/performance of the network used for the test - which was indeed performed on another server, so as not to affect the system deployment. Nonetheless, this failure helped to identify a defect in the system - i.e. the lack of error-feedback: "*This shouldn't happen. It didn't work but...you don't get any message of success or error. Also, it should not allow you to submit the request if something is wrong. This way it closes the form and you have to input everything again. If there is an input error, it should be displayed in the form. It is annoying*" (Participant 4). "*It is terrible that I've done a mistake and I've no idea what it was*" (Participant 5).

Task 5 - manual transaction (transfer energy from an EV to a utility company): this task has been performed easily by all participants and no particular feedback or suggestions for improvement emerged.

Task 6 - manual transaction (a resident withdraws cash): 4 out of 5 participants performed this task easily. Only one participant failed to do that due to a misunderstanding of the menu items. He indeed selected "Resident to Bank" from the list of transactions to perform since it appears before "Resident

Cash Withdraw": "*I didn't check all the items in the list...I thought that, since it involves Cash and Coin, it should have been Resident to Bank, so I didn't look further*" (Participant 1).

Except for a single system failure, the overall performance of the system resulted to be quite good. All participants reported to find the Admin UI clear and, when asked, they all say they would be able to use and explain it to someone else: "*I'd [be able to use and explain the system to someone else], despite this was the first time I've interacted with it. It might need to be slightly improved but definitely yes*" (Participant 1). All improvements suggested relates to the need of rewording some items - "*This 'Energy Value' is...one may think it is the production capacity, not the amount of energy he has. Should be something like 'energy available' or ' stored energy'*" (Participant 4) - and adding search tool like filters and sort options - "*there might be some tool to search for a given participant and/or sort the columns*" (Participant 3).

They also appreciated the EVs feature - "*EVs...ohhh...so the EVs are another part of the system...like Vehicle to Grid...that's cool!*" (Participant 4) - as well as the overall structure of the system and the logic behind it "*In general, it is clear. I mean, I've understood what it is about. Categories are clear...very clear*" (Participant 2).

Results from this test provided several valuable suggestions on how to significantly improve the system usability by rewording some items and making minor changes to the UI layout.

4.3 Quantitative assessment: end-user application and system backend

4.3.1 Methodology

To test the other system components - i.e. mobile app and system back-end - a small pilot was set up in Funchal (Madeira) involving the same sample that participated in the deployment of PSv1. Unfortunately, some of the monitoring systems installed in the participants' household got damaged and it was possible to recover and put in a working state only seven of them. One more smart meter was added to the system but, shortly after that, one of them failed and another started giving wrong production and consumption data. Due to these technical issues, only 6 households were included in the deployment to ensure reliability of the results.

The deployment lasted 3 weeks - from 7th to 28th of October 2019 - and after this date two malfunction smart-meters have been fixed so data are not included in the results but will be presented during the defence. Energy data was collected from the blockchain server and the ETMS server. Data regarding the interaction with the system and its performance was monitored and collected through the Fabric.io framework. A comparison between energy requirements of the old IOTA node and the new blockchain server is also made.

4.3.2 Results

During the experiment, all the energy transactions made using PSv2 were successfully recorded on both the ETMS database (in the transactions table) and the Hyperledger blockchain. In the new PSv2 system, the energy transactions are automatic, and payments and assets are updated automatically at the moment that the energy transaction is made. On the contrary, in the PSv1 system, this process requires users to perform some tasks. First of all, in order to perform a transaction and pay for it, users had to generate IOTA addresses. Secondly, the transaction was sent to the network for validation only after manual request from the user (i.e. by pressing the button “pagar a minhas despesas”). The result of such a low level of automation was that, at the end of the PSv1 deployment, none of the transactions performed was validated and recorded. Also in terms of transactions speed the new PSv2 system performed better compared to the previous one. With PSv1 the time needed to complete the query for new addresses generation, as well as to verify and validate a transaction, was longer than expected (up to several minutes) since both processes depended on the IOTA node state. As already mentioned, in the PSv2, payments are performed and balance of the assets updated at the moment the energy transaction is made.

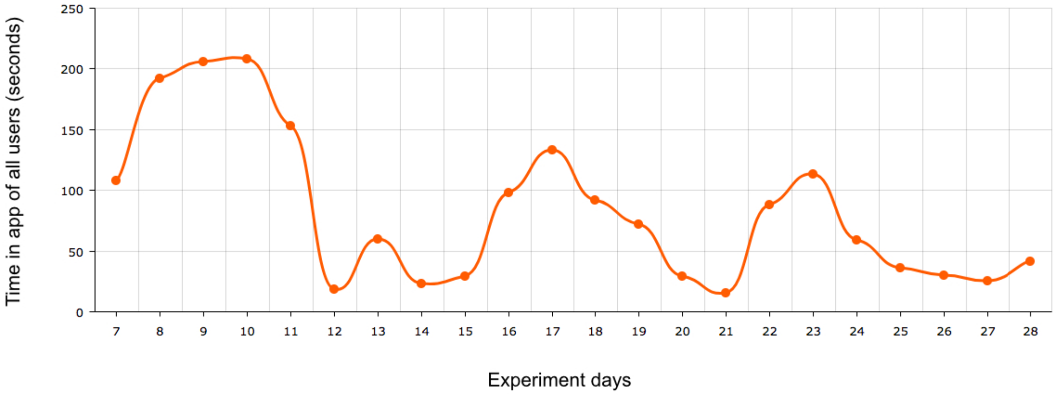


Figure 4.1: PSv2 android app time in app during the experiment

During the 3 weeks deployment, we counted 303 users’ sessions. For what concerns their distribution, as shown in figure 4.1 and 4.2, there was a peak in the first days after releasing the app, then the novelty effect passed and users’ interest in the application decreased. This result was expected, since users were already familiar with the application and no major changes were made on the UI except for the Hyperledger blockchain wallet. The same pattern was observed in the previous study.

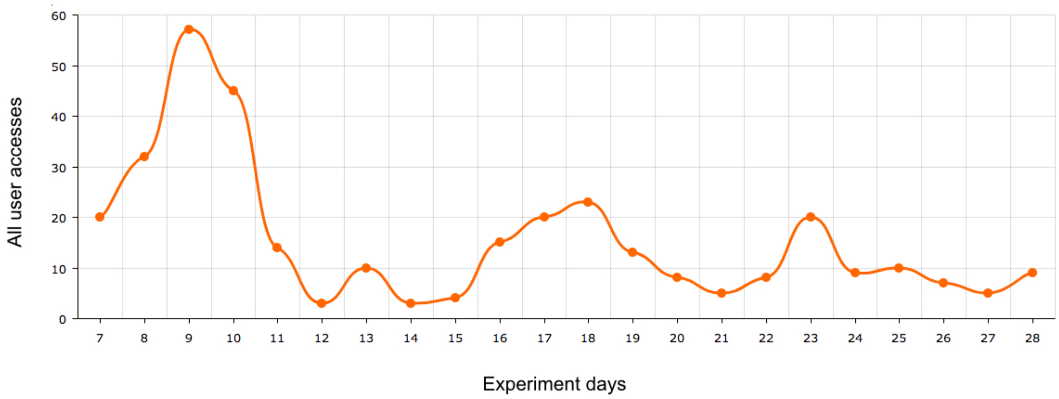


Figure 4.2: PSv2 android app accesses throughout the experiment

As shown in figure 4.3, the "home" was the most used feature. This is in line with the previous study and can be explained by the fact that it is the screen that opens by default when a user launches the app. Historical feedback was the second most accessed feature - once again, in line with results from the deployment of PSv1 - followed by the ranking. Unlike the previous study, where the "IOTA wallet" resulted to be the less popular feature, participants did access the Hyperledger blockchain wallet multiple times. This result suggests that, besides the initial curiosity (further demonstrated by the multiple accesses to the "token_help", which provides additional information about the wallet), people found it easier to use and understand compared to the previous IOTA wallet, and thus kept checking it.

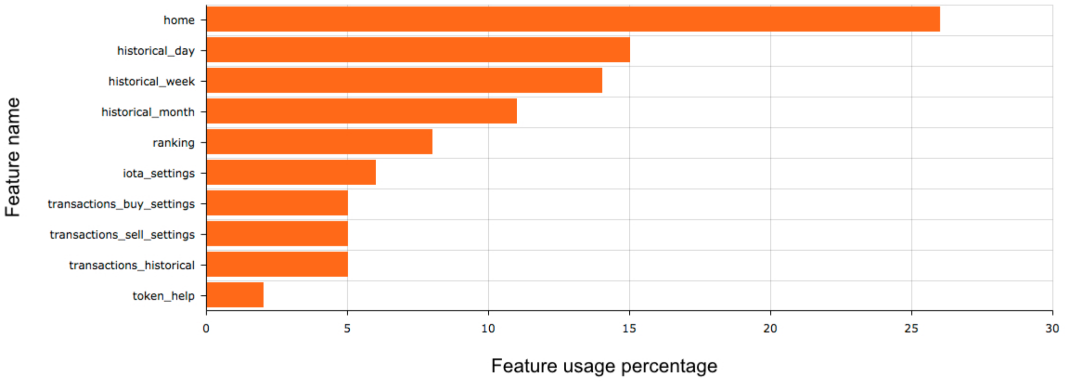
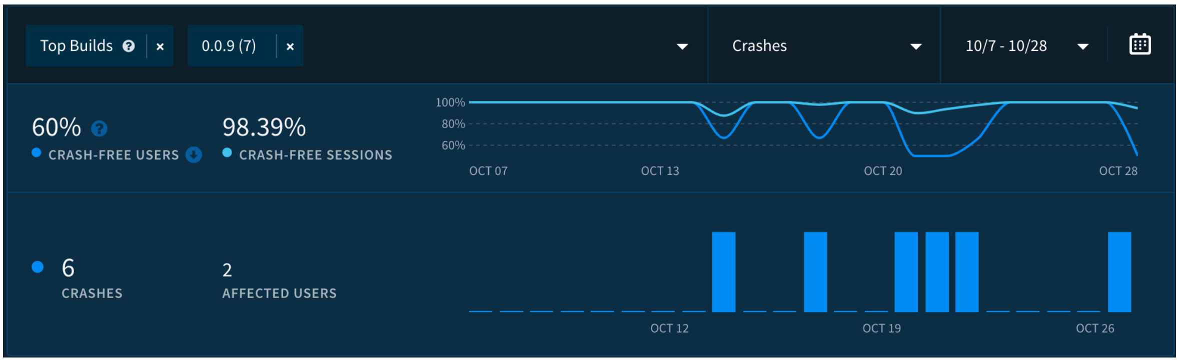


Figure 4.3: PSv2android app favorite features displayed based in the number of app accesses

In terms of system performance, we detected only one crash during the experiment (figure 4.4) which involved the "Ranking". In that case, the android app crashed unexpectedly during the download of a user images from the java server.



Top Builds ? × 0.0.9 (7) × Crashes 10/7 - 10/28

60% CRASH-FREE USERS

98.39% CRASH-FREE SESSIONS

6 CRASHES

2 AFFECTED USERS

Figure 4.4: PSv2 android app reported crash

During the deployment - that, as mentioned, involved 6 participants - a total of 207 energy transactions (table 4.1) were saved into both the ETMS server and the Hyperledger blockchain server, resulting in an average of almost 10 transactions per day. The transactions were generated by the ETMS that was built for PSv1 [5]. At the end of the experiment, the 207 energy transactions resulted in a total of 27.959 kWh exchanged within the community. Compared to the previous study, both the daily average transactions (10 vs 19) and the amount of energy exchanged (27.9kWh vs 45kWh) are lower. Nonetheless, it must be taken into account that in the deployment of PSv2 we had one third less participants (6 out of 9). In addition, a further aspect that could have influenced the result is the weather. Indeed, the previous study was conducted in September, while our deployment occurred in October and, during the three weeks from 7th to 28th of October 2019, we had several cloudy days.

Table 4.1: Transactions information during the three week study

	Number of Transactions	**Amount in €**	**Amount in kWh**
Min	3	0,004	0.023
Max	17	0,499	3.067
Average	9.9	0,216	1,331
Total	207	4,554	27.959

Concerning the system energy requirements, an attempt to compare the consumption of PSv1 and PSv2 was made. For the previous solution, a Ubuntu 16.04.4 server equipped with 6 dedicated cores processor and 8GB RAM was used to run the IOTA full node. The choice of running a private full node instead of relying on a public one was due to the need of having more control on it - i.e. to avoid interruptions and ensure it was always synchronized.

Since IOTA technology was abandoned, the Linux server has been replaced by a VPS running Ubuntu 16.04.6 LTS, equipped with 2 cores processor and 4GB RAM. This server is running all the Hyperledger Blockchain services (where the peers, Certificate Authority (CA) and Hyperledger orderer are implemented in Docker containers), the REST API and the Angular.js blockchain admin interface.

Although it's difficult to determine the server energy consumption based only on the number of CPU cores and the RAM deployed, an attempt to estimate such value for the two systems was made. Specifically, we made a small comparison between the consumption of two new and highly efficient server CPUs, one with 2 and the other with 6 cores. The RAM was not taken into account since it doesn't have a relevant impact on energy consumption.

The Intel Xeon D-1602 processor (with two cores running at 2.50GHz) and Intel Xeon D-1633N processor (with six cores running at 2.50GHz) were considered. Both released in the second quarter of 2019, revealed an impressive Thermal Design Power (TDP), respectively of 27W and 45W. Comparison information is available at https://www.intel.com/content/www/us/en/products/compare-products.html/processors?productIds=193686,193694

The two core CPU consumes 40% less energy than the six core CPU. In practice, this means that the first CPU has a daily energy consumption 0.648kWh and a yearly consumption of 236.52kWh, while the more powerful CPU has a 1.08kWh of daily energy consumption and a yearly energy consumption of 394.2kWh. In conclusion, after the comparison between the two similar processors, we can assume that the new blockchain implementation of the PSv2 is more efficient and consumes less energy than to the IOTA node used for the PSv1 implementation.

5

Conclusion

Contents

In this chapter, we make a small summary of the work done and the contributions that result from the development of this project. System limitations, scalability and future work are also discussed.

Blockchain is a disruptive technology that could innovate the energy sector since it has the ability to manage, store and certify data with no need for an external service provider. Blockchains could benefit the energy market by allowing the users (both consumers and prosumers) to sell and buy energy from each other at better rates, instead of selling the energy to traditional energy operators. Consequently, it has the power of challenging the current energy monopoly by making the market more competitive, while promoting the use of renewable energy sources.

A new version of the Power Share system, composed of the android app, ETMS and Hyperledger blockchain server, was successfully implemented and tested in a small pilot study run in Madeira from 7 to 28 October 2019. The IOTA cryptocurrency was replaced by a new smart contract blockchain system based on Hyperledger Fabric, bringing a more stable, energy-efficient system to perform energy transactions between residents. Also, the java app backend and the android app interface were upgraded to this new version, and the code was revised to improve overall stability and performance. Most of the current blockchain systems have high energy consumption for verification and validation of data [4]. To overcome this issue, PSv2 uses a permissioned blockchain system with a more efficient consensus algorithm, which results in lower energy consumption.

The new system also includes EVs, thus providing further opportunities to manage a decentralized energy system. “Advantages of decentralisation in this case include: the elimination of the need of a centrally managed EV charging infrastructure, fault tolerance, as well as elimination of price-setting and collusion between charging stations or transport providers.” [46]

At the end of the development, we have tested and evaluated our system by adopting both quantitative and qualitative approaches. This, in order to understand if it was clear and easy to use for both the Admin and end-users, as well as to assess its performance. Concerning users engagement with the Android application, which underwent minor changes, the results we obtained are similar to those from the previous study. On the contrary, we found evidence that the system efficiency, performance and automation of processes with blockchain service are significantly improved.

5.1 System Limitations

The current Power Share implementation consists of a small scale P2P energy trading community - i.e. 6 prosumers simulating a small microgrid. To assess the system scalability, a larger sample is needed. In addition, it should be pointed out that, since both the REST API endpoint and the web application - admin blockchain interface - are implemented on a single server, with all the blockchain components distributed on docker containers, the blockchain network as it is now is susceptible to Distributed Denial of Service Attack (DDOS) attacks - i.e.the service could be stopped causing transactions requests to be rejected by the blockchain service.

A further weak point of the system relates to the reliability of the data coming from the smart meters. With the current setting, a user can easily cheat the system simply by switching the production and consumption cables. This way, instead of buying energy, they could act as energy sellers and crediting them as energy producers. Since the monitoring system installed in the households of the participants in the study is part of the SMILE project demonstrator we could not act upon it. Nonetheless, it should be pointed out that, in a real-life scenario, tamper-proof seals should be implemented on the smart meters in order to ensure reliability of the data.

The end-user application, which is needed to join the energy community, has been developed only for Android based devices. This is a further limitation. To foster adoption of the system, a version of the application compatible with other OS platforms should be developed.

5.2 Scalability

The Hyperledger blockchain system we deployed consisted of only one blockchain peer, and all the necessary CA's and blockchain orderer were in docker containers. However, the system could be easily extended and used as a testbed to simulate a bigger scale energy transactive grid. The rationale behind the system is to allow for transactions between small microgrids (preventing hop and high transmission costs) but also to give Residents the opportunity to choose their energy supplier among all peers (figure 5.1).

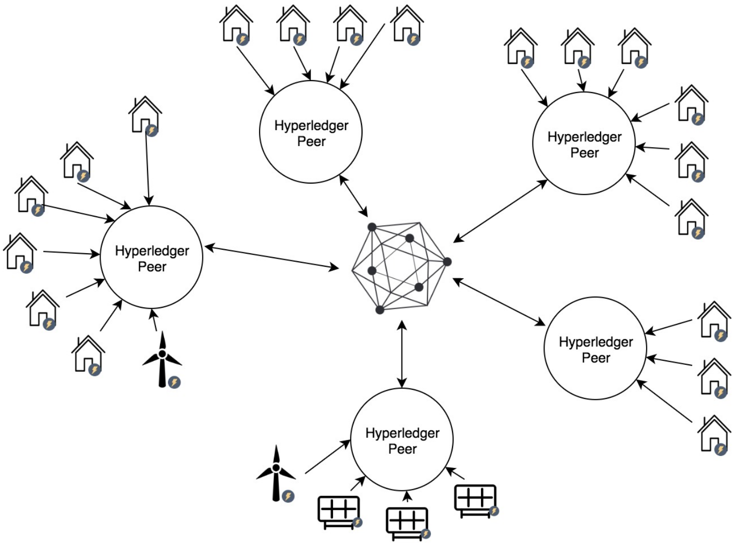


Figure 5.1: Representation of the Powershare system replicated to the Electrical Grid

The node peers would be those participating in the Hyperledger blockchain consensus and all the transactions on the grid would be recorded on the nodes and publicly available. This way, the system could be scaled up to cover the whole Madeira island, where these microgrids could be deployed by installing Hyperledger peers on small residents gathering, which would be then connected to the Hyperledger network. With the necessary adaptations, it could also be deployed to other locations and scales.

5.3 Future Work

The Power Share blockchain server was developed based on new technologies but there are still rooms for improvement from multiple points of view. The REST API and blockchain service were developed on Hyperledger Fabric v1.2, however, by the time of this writing, Hyperledger released version v1.4 which features significant improvements to the developer experience. The smart contract structure, Hyperledger network model and blockchain code could be updated to run on the newer Hyperledger versions.

The energy matching algorithm, that is currently running on the ETMS java server, could also be improved and moved to the blockchain system with the use of a smart contract energy bidding structure, thus further increasing the system automation. A further improvement relates to the algorithm itself, which so far does not take into account the losses that occur when the energy is injected into the grid and hop energy loss and cost to the destination.

Finally, the Android app could be upgraded by adding further network participants, specifically EVs and Utility Companies. In order to compare results from the deployment of PSv1 and PSv2, we decided to keep the same features of the previous version of the application. Nonetheless, by allowing end-user to interact and trade with further participants, we can provide them with further control over their energy suppliers and the source of the energy they are using, thus ultimately fostering the consumption of carbon free energy.

Bibliography

[1] E. Commission *et al.*, “Study on residential prosumers in the european energy union,” 2017.

[2] J. Froehlich, L. Findlater, and J. Landay, *The Design of Eco-Feedback Technology*, 2010. [Online]. Available: https://www.cs.umd.edu/{~}jonf/publications/Froehlich{_}TheDesignOfEcoFeedbackTechnology{_}CHI2010.pdf

[3] M. Mylrea and S. N. G. Gourisetti, “Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security,” in *Proceedings - 2017 Resilience Week, RWS 2017*. IEEE, sep 2017, pp. 18–23. [Online]. Available: http://ieeexplore.ieee.org/document/8088642/

[4] P. Global, “Blockchain-an Opportunity for Energy Producers and Consumers?” *PwC global power & utilities*, pp. 1–45, 2017. [Online]. Available: www.pwc.com/utilities

[5] T. Correia, “Power Share: Eco Feedback and Energy Trading System,” MSc dissertation, Universidade de Lisboa - Instituto Superior Técnico, Lisbon, 2018.

[6] S. Popov, “IOTA whitepaper v1.4.3,” pp. 1–28, 2018. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1{_}4{_}3.pdf

[7] V. H. Nguyen, Y. Besanger, Q. T. Tran, and M. T. Le, “On the applicability of distributed ledger architectures to peer-to-peer energy trading framework,” *arXiv preprint arXiv:1810.05541*, 2018.

[8] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: www.bitcoin.org

[9] M. G. Johnson, “Keyless signature infrastructure (ksi) overview,” *Guardtime Publication*, 2017.

[10] A. M. Antonopoulos, *Mastering Bitcoin 1. Edition*, 2010.

[11] M. Vukoli´cvukoli´c, “The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication,” Tech. Rep.

[12] C. Cachin and M. Vukoli´cvukoli´c, “Blockchain Consensus Protocols in the Wild,” Tech. Rep., 2017.

[13] M. Vukoli´cvukoli´c, "Rethinking Permissioned Blockchains." [Online]. Available: http://dx.doi.org/10.1145/3055518.3055526

[14] D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, p. 8, 2014.

[15] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2017, pp. 253–255.

[16] U. W. Chohan, "The double spending problem and cryptocurrencies," *Available at SSRN 3090174*, 2017.

[17] J. A. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications *," Tech. Rep., 2019.

[18] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 3–16.

[19] F. Saleh, "Blockchain without waste: Proof-of-stake," *Available at SSRN 3183935*, 2019.

[20] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," Tech. Rep., 2012. [Online]. Available: https://pdfs.semanticscholar.org/0db3/8d32069f3341d34c35085dc009a85ba13c13.pdf

[21] D. Larimer, "Delegated proof-of-stake (dpos)," *Bitshare whitepaper*, 2014.

[22] S. Mayer and E. O. S. Gmbh, "EOS Whitepaper," pp. 1–10, 2017. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[23] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[24] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," Tech. Rep., 1999.

[25] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2018.

[26] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53 019–53 033, 2018.

[27] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[28] N. Szabo, “Smart Contracts,” 1994. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[29] E. Androulaki Artem Barger Vita Bortnikov, C. Cachin Konstantinos Christidis Angelo De Caro David Enyeart, C. Ferris Gennady Laventman Yacov Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi Gari Singh Keith Smith Alessandro Sorniotti, C. Stathakopoulou Marko Vukolić Sharon Weed Cocco Jason Yellick, E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K.-n. Christidis, A. De Caro, D. Enyeart, C. Ferris, G.-n. Laventman, Y. Manevich, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, and S. Weed Cocco, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” *EuroSys*, vol. 18. [Online]. Available: https://doi.org/10.1145/3190508.3190538

[30] “Open source blockchain technologies.” [Online]. Available: https://www.hyperledger.org/

[31] J. Schlund, L. Ammon, and R. German, “Ethome: Open-source blockchain based energy community controller,” in *Proceedings of the Ninth International Conference on Future Energy Systems*, no. June. ACM, 2018, pp. 319–323. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3208903.3208929

[32] F. Imbault, M. Swiatek, R. de Beaufort, and R. Plana, “The green blockchain: Managing decentralized energy production and consumption,” in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. IEEE, jun 2017, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7977613/

[33] Hörchens and Ulrike, “Europe’s first blockchain project to stabilize the power grid launches: TenneT and sonnen expect results in 2018,” Tech. Rep., 2018. [Online]. Available: https://www.tennet.eu/fileadmin/user{_}upload/Company/News/German/Hoerchens/2017/20171102{_}PM-Start-Blockchain-Projekt-TenneT-sonnen{_}EN.pdf

[34] P. Danzi, M. Angjelichinoski, Č. Stefanović, and P. Popovski, “Distributed proportional-fairness control in microgrids via blockchain smart contracts,” in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, may 2017, pp. 45–51.

[35] D. Donnerer and S. Lacassagne, “Blockchain and energy transition-what challenges for cities?” 2018.

[36] M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. M. de Abril, and A. Nowé, “Nrgcoin: Virtual currency for trading of renewable energy in smart grids,” in *11th International conference on the European energy market (EEM14)*. IEEE, 2014, pp. 1–6.

[37] L03 Energy, “Blockchain based innovations to revolutionize how energy can be generated, stored, bought, sold and used, all at the local level,” 2018. [Online]. Available: https://lo3energy.com/

[38] “Sunchain.” [Online]. Available: https://www.sunchain.fr/

[39] “Produce one megawatt hour. get one free solarcoin.” [Online]. Available: https://solarcoin.org/

[40] “The first decentralized energy exchange platform powered by renewable.” [Online]. Available: https://pylon-network.org/

[41] L. Kalny, “Blockchain & energy: How the technology behind bitcoin can reinvent the way of doing business for utility providers,” Ph.D. dissertation, ESCP Europe, 2019.

[42] L. P. Johnson, A. Isam, N. Gogerty, and J. Zitoli, “Connecting the blockchain to the sun to save the planet,” 2015.

[43] “Introducing eosio 2: Enhancing performance, improving security, and new developer tools,” Oct 2019. [Online]. Available: https://eos.io/news/introducing-eosio-2/

[44] A. H. Lone and R. N. Mir, “Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer,” *Digital Investigation*, vol. 28, pp. 44–55, 2019.

[45] C. Tiefenau, E. von Zezschwitz, M. Häring, K. Krombholz, and M. Smith, “A usability evaluation of let’s encrypt and certbot: Usable security done right,” 2019.

[46] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, “Blockchain technology in the energy sector: A systematic review of challenges and opportunities,” *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

A

Power Share Business Model

The CTO file that contains all the definitions for the business model of the Power Share Hyperledger Blockchain Service .

Listing A.1: Hyperledger business model - Model.cto

```
/**
 * Decentalized Smile energy network definition
 */
namespace org.smile.energy.network

/**
 * Type of EV
 */
enum ElectricVehicleType {
  o BEV
  o PHEV
  o HEV
  o SCOOTER
  o BIKE
}

/**
 * Type of Participant
 */
enum OwnerEntity {
  o Resident
  o Bank
  o UtilityCompany
  o ElectricVehicle
}

participant Resident identified by residentID {
    o String residentID
    o String firstName
    o String lastName
    o String email
    o Double buyRate default = 0.1629
    o Double sellRate default = 0.1629
    --> Coins coins
    --> Cash cash
    --> Energy energy
}

participant ElectricVehicle identified by evID {
    o String evID
    o ElectricVehicleType type
    o String brand
    o String model
    o String ownerID
    o Double buyRate default = 0.1629
    o Double sellRate default = 0.1629
    --> Coins coins
    --> Energy energy
}
```

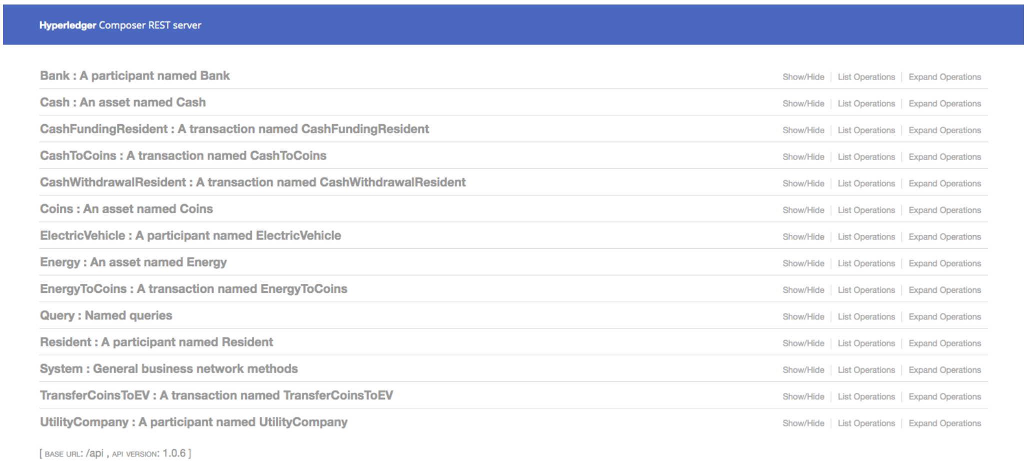
```

participant Bank identified by bankID {
    o String bankID
    o String name
    --> Coins coins
    --> Cash cash
}

participant UtilityCompany identified by utilityID {
    o String utilityID
    o String name
    --> Coins coins
    --> Energy energy
}

asset Coins identified by coinsID {
    o String coinsID
    o Double value
    o String ownerID
    o OwnerEntity ownerEntity

}

asset Energy identified by energyID {
    o String energyID
    o String units
    o Double value
    o String ownerID
    o OwnerEntity ownerEntity
}

asset Cash identified by cashID {
    o String cashID
    o String currency
    o Double value
    o String ownerID
    o OwnerEntity ownerEntity
}
```

```


transaction EnergyToCoins {
    o Double energyRate
    o Double energyValue
    --> Coins coinsInc
    --> Coins coinsDec
    --> Energy energyInc
    --> Energy energyDec
}

transaction CashToCoins {
    o Double cashRate
    o Double cashValue
    --> Coins coinsInc
    --> Coins coinsDec
    --> Cash cashInc
    --> Cash cashDec
}

transaction TransferCoinsToEV {
    o Double coinsValue
    --> Coins coinsInc
    --> Coins coinsDec
}

transaction CashFundingResident {
    o Double cashValue
    --> Cash cashInc

}

transaction CashWithdrawalResident {
    o Double cashValue
    --> Cash cashDec
}
```

B

API Documentation

The documentation and service explorer of the API provided to interact with the Hyperledger Blockchain could be found at https://ruben.m-iti.org:3000/. The updated documentation of the ETMS service could be found at https://documenter.getpostman.com/view/3031217/SW12zcoZ.



Hyperledger Composer REST server

Bank : A participant named Bank

Cash : An asset named Cash

CashFundingResident : A transaction named CashFundingResident

CashToCoins : A transaction named CashToCoins

CashWithdrawalResident : A transaction named CashWithdrawalResident

Coins : An asset named Coins

ElectricVehicle : A participant named ElectricVehicle

Energy : An asset named Energy

EnergyToCoins : A transaction named EnergyToCoins

Query : Named queries

Resident : A participant named Resident

System : General business network methods

TransferCoinsToEV : A transaction named TransferCoinsToEV

UtilityCompany : A participant named UtilityCompany

[BASE URL: /api , API VERSION: 1.0.6]

Figure B.1: Hyperledger blockchain endpoint explorer