



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO INDUSTRIAL

## **Blockchain Technology Application to Energy Communities**

**Enrique Gómez Limia**

Supervisors: José Pablo Chaves Ávila, Daniel Fernández Lestón

Madrid

July 2018



## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO**

### ***1º. Declaración de la autoría y acreditación de la misma.***

El autor D. **Enrique Gómez Limia** DECLARA ser el titular de los derechos de propiedad intelectual de la obra: “**Blockchain technology to energy applications**”, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### ***2º. Objeto y fines de la cesión.***

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### ***3º. Condiciones de la cesión y acceso***

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### ***4º. Derechos del autor.***

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

### ***5º. Deberes del autor.***

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

**6º. Fines y funcionamiento del Repositorio Institucional.**

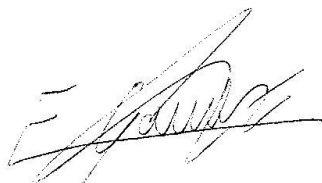
La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 21 de Junio de 2018

**ACEPTA**

Fdo: Enrique Gómez Limia



Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
**“Blockchain technology application to energy communities”**  
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2º de Máster es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro,  
ni total ni parcialmente y la información que ha sido tomada  
de otros documentos está debidamente referenciada.

Fdo.: Enrique Gómez Limia

Fecha: 21 / 06 / 2018



Autorizada la entrega del proyecto

**EL DIRECTOR DEL PROYECTO**

Fdo.: José Pablo Chaves Ávila Fecha: 22/06/ 2018

*José Pablo Chaves Ávila.*



# APLICACIÓN DE LA TECNOLOGÍA BLOCKCHAIN EN COMUNIDADES ENERGÉTICAS

**Autor: Gómez Limia, Enrique**

Directores: Chaves Ávila, José Pablo y Fernández Lestón, Daniel

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

## RESUMEN DEL PROYECTO

### Introducción

En la última década, el sector eléctrico está experimentando grandes cambios. El calentamiento global está fomentando la introducción de la energía renovable, hay un aumento de la demanda global, la digitalización está aumentando, los consumidores se están volviendo más activos y cada vez más recursos energéticos se están introduciendo a nivel distribuido. Además, la actitud de los consumidores se está orientando hacia un mayor interés por la obtención de servicios que de productos y la economía colaborativa está en su mayor esplendor. Paralelamente a este escenario, Blockchain, la tecnología subyacente detrás de Bitcoin y todas las demás criptomonedas, aparece con el objetivo de transformar los negocios basados en transacciones digitales.

Blockchain permite la garantía de la unicidad de un activo digital sin la necesidad de un tercero de confianza. En su lugar, un sistema basado en incentivos económicos integrados en la propia tecnología alienta a un grupo de participantes desconocidos a dedicar recursos para garantizar la no duplicidad

de los bienes digitales. Este nuevo paradigma permite la ejecución de transacciones y el intercambio digital de valor asegurando la ausencia de doble gasto.

Por lo tanto, la integración de Blockchain en la industria energética puede dar lugar a la automatización de algunos de sus procesos con la consiguiente mejora de la eficiencia del sistema. Debido a la estructura del mercado y al marco regulatorio, la mayor parte del trabajo se está haciendo a nivel local con el fin de interactuar lo menos posible con la red central y chocar con obstáculos regulatorios. En esta línea, Blockchain puede usarse para integrar transacciones de energía P2P, crear mercados locales de electricidad y administrar comunidades locales de energía. Los proyectos más importantes que están trabajando en esta área son TransActive Grid con su micro red gobernada por Blockchain en Brooklyn y Power Ledger en Australia. En España, Pylon Network es la más importante, durante el último año está teniendo un gran desarrollo y está firmando algunos contratos de colaboración con las cooperativas de energía españolas.

El objetivo principal del proyecto es idear y construir una herramienta que sea capaz de gestionar de forma autónoma una comunidad energética local. Debe permitir a sus miembros invertir en tecnologías de autoconsumo compartido y registrar automáticamente las mediciones de energía con su correspondiente seguimiento financiero en forma de transacciones económicas entre usuarios.

Otros objetivos secundarios son los de comprender mejor la tecnología y extraer las principales ventajas y limitaciones que tiene esta para penetrar en el sector eléctrico, así como revisar el estado del arte realizado hasta la fecha en el campo de estudio.

### Metodología

La comunidad energética tomada en consideración para la elaboración del prototipo se muestra en la Figura I.

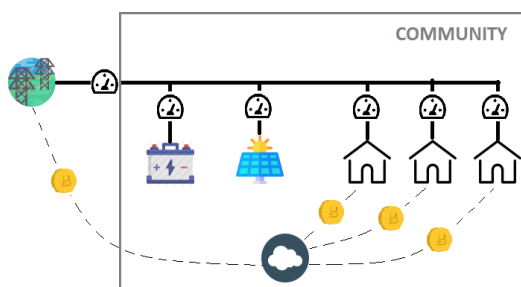


Figura I. Esquema de Comunidad Energética

La comunidad energética está formada por un grupo de demandas y tecnologías de autoconsumo compartidas, en las que los miembros pueden invertir, todas ellas con su medidor inteligente conectado a Blockchain. Solo existe un punto de

conexión con la red externa. También hay un agente llamado Banco Comunitario u Owner que se encarga de instalar las tecnologías de autoconsumo, emitir tokens de participación y recaudar y distribuir pagos entre los miembros. Esta función puede desempeñarla una comercializadora, una cooperativa o una asociación formada dentro de la comunidad.

La plataforma creada tiene dos partes bien diferenciadas:

En primer lugar, una herramienta de optimización, ejecutada por una GUI de Matlab, llamada OBAMA. Se utiliza para dimensionar de manera óptima las tecnologías de autoconsumo que se instalarán en la comunidad energética. Esto se realiza tomando como inputs características de la comunidad como el número de participantes, sus curvas de demanda, las curvas de radiación en la región y la tarifa contratada. Esta herramienta también genera el funcionamiento óptimo de las tecnologías de autoconsumo y la potencia óptima que la comunidad debe contratar. OBAMA es utilizada como herramienta de previsión para que la comunidad energética compre o venda la electricidad requerida en el mercado eléctrico.

En segundo lugar, se programó una plataforma Blockchain en forma de aplicación distribuida, que incluye todas sus partes (JavaScript, HTML, CSS y Smart Contract en Solidity). Esta plataforma es responsable de emitir



tokens de participación para la financiación de las tecnologías a instalar. Asimismo, registra todas las mediciones de energía entrante de cada uno de los medidores inteligentes y los precios de mercado eléctrico. A partir de ahí, asigna los haberes o deberes económicos a cada miembro y liquida las correspondientes transacciones.

La plataforma opera bajo la utilización de dos tokens. El EC coin (Energy Community) es el utility token de la aplicación, los usuarios hacen uso de él para pagar su factura de electricidad y para poder invertir en autoconsumo compartido. El EC coin se puede intercambiar por dinero fiduciario u otras criptomonedas en casas de cambio. Los tokens SCP (Self Consumption Participation) son tokens que representan participaciones en las tecnologías de autoconsumo instaladas. Los SCP son emitidos por el Banco Comunitario para financiar las tecnologías. Se comercializan en un mercado interno. Brindan beneficios a sus propietarios sobre la energía generada en forma de EC coins.

La Figura II muestra los flujos de los diferentes tokens entre los agentes participantes dentro de la comunidad energética.

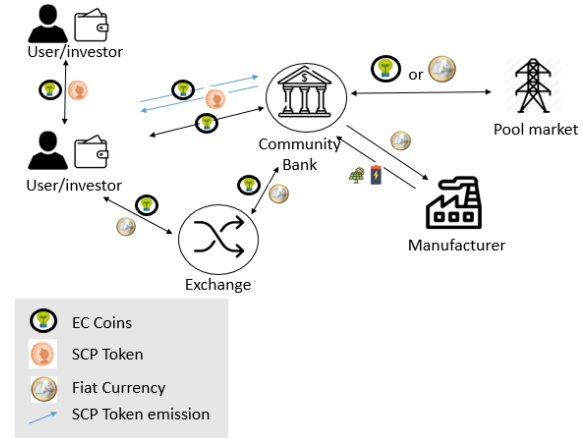


Figura II. Mapa de valor de la comunidad energética

Para el desarrollo del prototipo se utilizó una blockchain privada ejecutada en local y que funciona en la máquina virtual de Ethereum.

### Caso de estudio y resultados

Para las simulaciones y testeo del prototipo, se han utilizado las curvas de demanda reales de un grupo de 3 casas unifamiliares en Valencia (España) del año 2017. En cuanto a las tecnologías de autoconsumo a instalar, se consideraron un panel fotovoltaico y una batería. La curva de precios se obtiene sumando los precios de mercado para cada hora del año, la tarifa de acceso de la comunidad y los impuestos de la red.

Después de analizar el caso y simular con OBAMA, se calculan los tamaños óptimos de PV y baterías en los que la comunidad debería invertir. Estos se muestran junto con sus precios de inversión en la Tabla I. Para realizar un análisis CBA, la vida del proyecto se establece a 20 años teniendo que sustituir la batería por otra igual en el año 10.

Tecnología y dimensión	Coste de inversión
Batería: <b>6.4 kWh</b> con un inversor de 3 kW	Año 0: 3,700 €
	Año 10: 1,840 €
PV panel: <b>4 kWp</b>	Año 0: 4,000 €

Tabla I. Tamaños óptimos y costes de inversión de las tecnologías de autoconsumo a instalar

OBAMA también calcula la operación óptima de los recursos energéticos. A partir de esto, se puede calcular el ahorro anual esperado proporcionado por las tecnologías. La Figura III muestra los flujos de caja de la comunidad energética que servirán para mostrar a los inversores la rentabilidad que los SCP tokens pueden otorgarles.

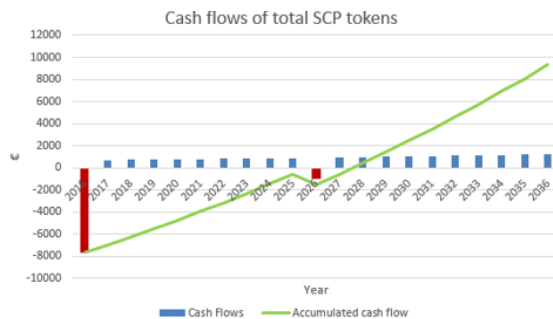


Figura III. Flujos de caja de la totalidad de SCP tokens de la comunidad energética

Para simplificar, las previsiones se consideran perfectas y, por lo tanto, la medición y los precios entrantes cada hora durante un año completo se derivan directamente de OBAMA. Estos sirven, por tanto, como datos de entrada a la plataforma Blockchain que se ejecuta en tiempo real.

Para el caso de estudio, se emiten 770 SCP tokens con valor de 10 € cada uno para cubrir la inversión inicial de 7.700 €. Como se muestra en la Tabla II, los usuarios compran la siguiente cantidad de SCP tokens.

Usuario 1	77 SCP	10 %
Usuario 2	539 SCP	70 %
Usuario 3	154 SCP	20%

Tabla II. Participaciones en tecnologías de autoconsumo de cada usuario

La Dapp (App Distribuida) elaborada fue testada para probar su correcto funcionamiento. Como ejemplo, la Figura IV muestra las transacciones económicas por hora y el saldo acumulado en EC del usuario 2 durante las primeras 24 horas de simulación (1 de enero de 2017).

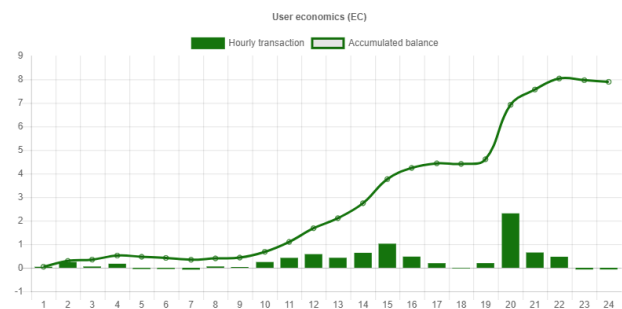


Figura IV. Transacciones de EC coins y balance acumulado del Usuario 2 durante las primeras 24 horas

Asimismo se calculó el saldo final de cada usuario al final del día para demostrar que se alcanzó el equilibrio. La Tabla III muestra los resultados.

Usuario 1	-8.796 EC
Usuario 2	7.896 EC
Usuario 3	-0.796 EC
DSO	1.696 EC
<b>Total</b>	<b>0 EC</b>

Tabla III. Balances acumulados en la hora 24

Finalmente, la tabla IV muestra los ahorros en la factura anual de electricidad para cada uno de los usuarios considerando que mantienen el % de SCP tokens durante todo el año.

Usuario 1 (10% SCP)	70 € (33%)
Usuario 2 (70% SCP)	491 € (124%)
Usuario 3 (20% SCP)	140 € (22%)

Tabla IV. Ahorro en la factura eléctrica de cada usuario

El usuario 1 puede ahorrar el 33% de su factura con solo comprar un 10% de los tokens. Por otro lado, el Usuario 2 cubre el total de sus gastos de factura de electricidad e incluso gana algo de dinero.

### Conclusión

En tiempos de transición energética hacia un mercado eléctrico más renovable y descentralizado, esta tesis ha desarrollado una plataforma Blockchain que mide los flujos de energía y ejecuta las transacciones económicas en tiempo real de una comunidad energética, formación que está ganando en popularidad.

Esta plataforma optimiza las dimensiones de las tecnologías de generación a instalar, permite a los participantes invertir en esas tecnologías comprando tokens virtuales, lo cual aumenta su liquidez. Mantiene un registro inmutable de las medidas que provienen de los medidores inteligentes y las traduce en transacciones económicas. En definitiva, con esta plataforma, los usuarios pueden ser más conscientes y participativos de los recursos energéticos de la comunidad de forma transparente y confiable, pueden invertir en energía renovable de una manera más simple y pueden contribuir social y económicamente a la transición energética hacia un planeta más limpio.

Como trabajo futuro, la plataforma debe mejorarse para que pueda aplicarse a cualquier comunidad con cualquier cantidad de miembros. Además, los algoritmos de pronóstico, así como los incentivos para la respuesta a la demanda deben incluirse para lograr un prototipo más realista y completo. También se deja como trabajo futuro la elaboración de un plan de negocio del prototipo y el análisis de su integración en los marcos regulatorios de diferentes países.

Finalmente, se espera que este prototipo puede servir como punto de partida para futura investigación académica y empresarial y la elaboración de un proyecto piloto que pueda resultar en el desarrollo de una plataforma profesional y comercial.

## BLOCKCHAIN TECHNOLOGY APPLICATION IN ENERGY COMMUNITIES

**Author: Gómez Limia, Enrique**

Supervisors: Chaves Ávila, José Pablo and Fernández Lestón, Daniel

Entity in Collaboration: ICAI – Universidad Pontificia Comillas

### SUMMARY OF THE PROJECT

#### Introduction

The electricity sector is undergoing big changes in the last decade. Global warming is fostering the introduction of renewable energy, there is being an increase of the overall demand, digitalization is increasing, consumers are becoming more active and electricity services are being introduced at a distributed level. In addition, consumers' attitude is shaping towards a higher interest in services than in products and the sharing economy is at its greatest splendor. Parallel to this scenario, Blockchain, the underlying technology behind Bitcoin and every other cryptocurrency appears with the aim of disrupting digital transactions based businesses.

Blockchain allows the guarantee of the uniqueness of a digital asset without the need of a trusted third party. In its place, a system based on economic incentives embedded in the technology encourages a pool of unknown participants to dedicate resources to ensure the non-duplicity of digital assets. This new paradigm permits the execution of

transactions and the digitally exchange of value assuring the absence of double spending.

Therefore, the integration of Blockchain in the energy industry can lead to the automation of some of its processes with the consequent improvement of the efficiency of the system. Due to the structure of the market and the regulation framework, most of the work is being done at a local level with projects that interact with the central grid as less as possible so that to prevent facing too many regulative drawbacks. In this line, Blockchain can be used to integrate energy P2P transactions, create local electricity markets and manage local energy communities. The most important projects that are working on the topic are TransActive Grid with its Blockchain governed microgrid in Brooklyn and Power Ledger in Australia. In Spain, Pylon Network is the most important one and is making good progress over the last year signing some collaboration contracts with Spanish energy cooperatives.

The main objective of the project is to ideate and build a tool that is able to autonomously manage a local energy community. It should allow its members to invest in shared self-consumption technologies and automatically register energy measurements with the corresponding financial tracking in form of economic transactions between users.

Other secondary objectives were to understand better the technology and draw out the main advantages and limitations it has to penetrate the electricity sector, as well as revise the progress to date done in the topic.

### Methodology

The energy community taken into consideration to build the prototype can be seen in Figure I.

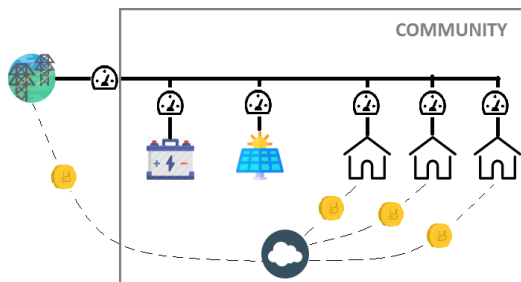


Figure I. Energy Community schema

It is formed by a group of demands and shared self-consumption technologies in which members can invest, all of them with their Blockchain connected smart meter. There is only one point of connection with the external grid. There is also a figure called Local Community Bank or Owner that is in charge of installing the self-consumption

technologies, emitting participation tokens and collecting and distributing payments between members. This role can be played by a retailer, a cooperative or an association formed inside the community.

The platform built has two well distinctive parts:

First, an optimization tool, run by a Matlab GUI, called **OBAMA**. It is used for sizing the optimal dimensions of the self-consumption technologies to be installed in the energy community. This is done taking as inputs some community characteristics such as the number of participants, their demand curves, radiation curves in the region and the tariff contracted. This tool outputs as well the optimal operation of the self-consumption technologies and the optimal power to be contracted by the community. It serves as the forecasting tool for the Energy community to buy or sell the required electricity at the pool market.

Second, a Blockchain platform in form of a Distributed app was programmed, including all its parts (JavaScript, HTML, CSS and Solidity smart contract). This platform is responsible for emitting participation tokens so that investors can buy them and the technologies to be installed are financed. It also keeps record of all the incoming energy measurements from each of the smart meters and the electricity market prices. From that, it assigns economic

debts and earnings to each member and settles the corresponding transactions.

For the operation of the platform it was implemented a dual token based system. The EC (Energy Community) coin is the utility token of the application, users will need them to pay for their electricity bill and to be able to invest in shared self-generation. It can be bought or sold for fiat currency or other cryptocurrencies in exchange houses. The SCP (Self Consumption Participation) tokens are asset based tokens, they represent shares in the self-consumption technologies installed and they are emitted by the Community Bank to finance the technologies. They are traded in an internal market. They provide benefits to their owners on the energy generated in form of EC coins.

Figure II shows the flows of the different tokens between the participant agents within the energy community.

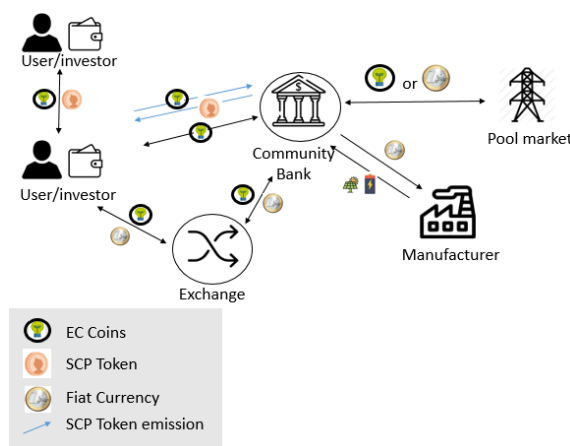


Figure II. Value flow map of the community

For the development of the prototype it was used a private blockchain running

locally and based in Ethereum Virtual Machine.

### Case study and results

For the prototype and testing simulations run, a group of 3 single-family houses in Valencia (Spain) and their demand curves for year 2017 were used. Regarding the self-consumption technologies to be installed, a PV panel and a battery were considered. The curve of prices is derived by the addition of the market prices for each hour of the year, the access tariff of the community and the network taxes.

After analyzing the case and simulating with OBAMA, the optimal sizes of PV and batteries in which the community should invest are calculated. They are shown together with their investment prices in Table I. In order to make a CBA analysis, the life of the project is established to 20 years having to buy and substitute the battery in year 10.

Technology and size	Investment cost
Battery: <b>6.4 kWh</b> with an inverter of 3 kW	Year 0: 3,700 €
	Year 10: 1,840 €
PV panel: <b>4 kWp</b>	Year 0: 4,000 €

Table I. Optimal sizes and investment costs of the technologies to be installed

OBAMA also outputs the optimal operation of the energy. From this, the expected savings provided by the technologies can be calculated. Figure III shows the cash flows of the energy

community that will serve to show the investors the profitability that the SCP tokens can give them.

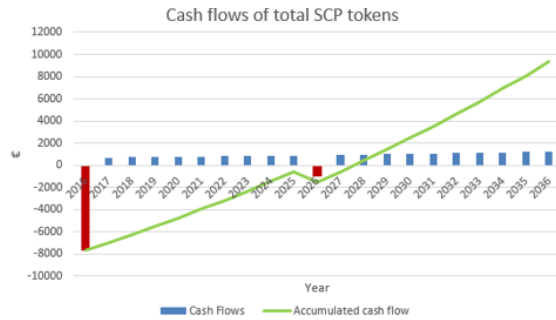


Figure III. Cash flows of the total amount of SCP tokens of the Community

For simplicity, the forecasts are considered to be perfect and therefore the incoming measurement and prices at each hour for one full year are directly derived from OBAMA. They will serve as input to the Blockchain platform that is run in real time.

For the case of study, 770 SCP tokens with value of 10 € each are emitted to cover the initial investment of 7,700 €. As shown in Table II, users buy the following amount of SCP tokens

User 1	77 SCP	10 %
User 2	539 SCP	70 %
User 3	154 SCP	20%

Table II. Users' participation in self-consumption technologies

The Dapp built was tested to prove that it worked properly. As example, Figure IV shows the economic hourly transactions and accumulated balance in EC of user 2 for the first 24 hours of simulation (January the 1<sup>st</sup> 2017).

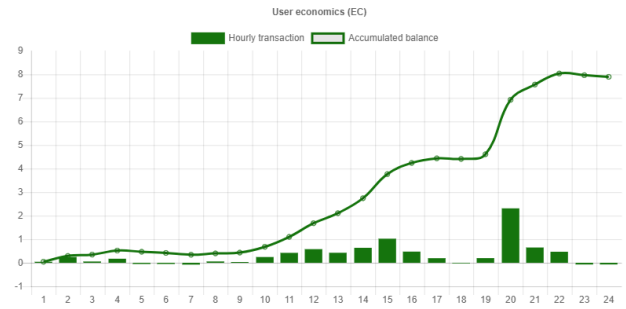


Figure IV. EC transactions and accumulated balance of User 2 for 24 hours

For verification purposes, it was also calculated the final balance of each user at the end of the day to prove that equilibrium was reached. Table III shows the results.

User 1	-8.796 EC
User 2	7.896 EC
User 3	-0.796 EC
DSO	1.696 EC
<b>Total</b>	<b>0 EC</b>

Table III. Accumulated balance in hour 24

Finally, in table IV it is shown the savings in the annual electricity bill for each of the users if they maintained the % of SCP tokens during the whole year.

User 1 (10% SCP)	70 € (33%)
User 2 (70% SCP)	491 € (124%)
User 3 (20% SCP)	140 € (22%)

Table IV. Savings in the electricity bill for each of the users of the community

Evidently, the higher the % of shares in self-consumption, the higher the savings the user will get. User 1 is able to save

33% of his bill with only buying a 10% of the tokens, his electricity bill is low and therefore he does not need to buy a lot of SCP tokens to cover its expenses. On the other hand, User 2 is able to surpass the total of his electricity bill expenses and even earn some money out of the generation of electricity.

### **Conclusion**

In times of energy transition into a more renewable and decentralized electricity market, this thesis has developed a Blockchain platform that measures the energy flows and executes the economic transactions in real time of an energy community, formation that is gaining in popularity.

This platform is able to optimize the dimensions of the generation technologies to be installed, it allows participants to invest in those technologies by buying digital tokens that can be exchanged or sold at any time, rising their liquidity. It keeps an immutable record of the measurements coming from smart meters and it couples them to economic transactions that are as well automatically executed between members. Ultimately, with this platform, users can become more aware and participative in electricity services inside the community in a transparent and trustworthy way, they can invest in renewable energy in a simpler manner and they can socially and economically contribute to the energy transition towards a cleaner planet.

As future work, the platform has to be improved so that it becomes scalable and can be applied to any community with any number of members. In addition, forecasting algorithms as well as incentives for demand response should be included to achieve a more realistic and complete prototype. It is also left as future work the elaboration of a business plan of the prototype and the analysis of its integration in the regulation frameworks of different countries.

Finally, the delivery of this prototype can serve as a starting point for future academic and company research and the elaboration of a pilot project that could result in the development of a professional and commercial platform.







## *Acknowledgements*

The completion of this thesis would not have been possible without the support of many people along the way.

I first want to thank my two supervisors, José Pablo and Daniel, you have supported me and have dedicated a big amount of time to my work. Without your guidance and enlightening, the results achieved would have never been possibly reached. You made easy and simple what looked complex for me in the beginning.

Thanks to the team behind the elaboration of OBAMA tool, José María, Enrique and Lukas for letting me use it and being so opened in case any problem arose. In this same line, thanks to Alezeia and the team of GNF where I learned how to use this tool and I acquired a lot of knowledge about self-consumption technologies and more specific about batteries.

I also want to thank Daniel Díez and his blockchain team in UST global for their guidance and important hints for the ideation and programming of the platform developed in this thesis.

I am also appreciative of the Blotick team and our mentor Moisés. I have learned a lot about Blockchain and its fundamentals thanks to forming part of this awesome team.

Finally and most important I want to thank every person that is close to me and that have shaped me as the person I am throughout the last years. Thanks to my Delicias group of friends that I have since I was a kid. Thanks to all the friends I have made during these 6 years of university and thanks to my girlfriend, Marusela, you have always understood me and known how to help me in every single thing. Finally, I am mostly appreciative of my family for their unconditional support. Thank you Mom and Dad for giving me the chance to study in this university and always have been there for me. Thank you Dad for your involvement and help with the thesis, you have always served as inspiration to me. Thanks Guille and Celia for your continuous support and making life easier at home being so good to me.





ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

## **Blockchain Technology application to Energy Communities**

**Enrique Gómez Limia**

Supervisors: José Pablo Chaves Ávila, Daniel Fernández Lestón



## Contents

<b>Chapter 1</b>	<b>Introduction .....</b>	<b>7</b>
<b>Chapter 2</b>	<b>Blockchain Technology.....</b>	<b>11</b>
2.1	<b>Blockchain and its origin.....</b>	<b>11</b>
2.2	<b>Explanation of blockchain through its main terms .....</b>	<b>13</b>
2.3	<b>Types of consensus protocols .....</b>	<b>16</b>
2.4	<b>Types of Blockchains .....</b>	<b>18</b>
2.5	<b>Blockchain stack.....</b>	<b>20</b>
2.6	<b>Sectors to be disrupted by Blockchain.....</b>	<b>22</b>
<b>Chapter 3</b>	<b>Evolution of the power industry .....</b>	<b>25</b>
3.1	<b>structure of the electricity grid and market .....</b>	<b>25</b>
3.2	<b>Trends.....</b>	<b>26</b>
3.2.1	Climate change.....	26
3.2.2	Decentralization.....	28
3.2.3	Digitalization.....	31
3.3	<b>Local energy communities.....</b>	<b>32</b>
<b>Chapter 4</b>	<b>State of the art: Blockchain for P2P energy trading and Energy Communities 35</b>	
<b>Chapter 5</b>	<b>Local energy community governed by a Blockchain platform ... 41</b>	
5.1	<b>Description of the proposed Blockchain governed energy community.....</b>	<b>41</b>
5.1.1	Components of the community.....	41
5.1.2	Features and main parts of the platform.....	43
5.1.2.1	Optimization and forecasting tools.....	43
5.1.2.2	Blockchain platform for the energy community.....	44
5.1.3	Functioning of the platform.....	47
5.2	<b>Main advantages of a Blockchain energy community .....</b>	<b>52</b>
5.3	<b>Main limitations and risks of using Blockchain for energy applications .....</b>	<b>53</b>
5.4	<b>Suitable types of Blockchain and consensus .....</b>	<b>55</b>
5.5	<b>Parameters and Hypotheses for the developed prototype.....</b>	<b>56</b>
<b>Chapter 6</b>	<b>Optimization simulation tool: OBAMA.....</b>	<b>61</b>
6.1	<b>OBAMA.....</b>	<b>61</b>
6.2	<b>Methodology .....</b>	<b>62</b>
6.2.1	Demand curve analysis .....	62
6.2.2	Prices curve analysis.....	63
6.2.3	Radiation curve .....	63
6.2.4	PV and batteries preliminary sizing.....	64
6.2.5	Simulations with OBAMA.....	65
6.2.6	Costs and benefits analysis.....	65
<b>Chapter 7</b>	<b>Description of the Dapp created.....</b>	<b>67</b>
7.1	<b>Programming Tools used for the development.....</b>	<b>67</b>
7.1.1	NPM (Node Package Manager).....	67
7.1.2	Solidity.....	67

7.1.3 HTML, JavaScript and CSS .....	69
7.1.4 Ganache .....	69
7.1.5 Web3 and MetaMask .....	70
7.1.6 Truffle .....	72
<b>7.2 Architecture of the Dapp.....</b>	<b>73</b>
7.2.1 Upper level folder .....	75
7.2.1.1 Package.json .....	76
7.2.1.2 Truffle.js .....	76
7.2.1.3 Webpack.Config.js .....	76
7.2.2 Contracts folder .....	77
7.2.2.1 Migrations.sol .....	77
7.2.2.2 energyCommunity.sol.....	77
7.2.3 Build folder .....	81
7.2.4 Migrations folder .....	82
7.2.4.1 Initial migration.....	82
7.2.4.2 Deploy contracts migration .....	82
7.2.5 App folder .....	83
7.2.5.1 Data.json .....	83
7.2.5.2 Index.html and app.CSS .....	84
7.2.5.3 app.js .....	89
7.2.6 Node_modules folder.....	91
<b>7.3 Limitations of the Dapp .....</b>	<b>91</b>
<b>Chapter 8 Case study, Validation of the prototype and results.....</b>	<b>93</b>
8.1 <b>OBAMA simulations .....</b>	<b>93</b>
8.2 <b>Initialization of the Dapp .....</b>	<b>102</b>
8.3 <b>Simulations with the Dapp .....</b>	<b>103</b>
8.4 <b>economic analysis .....</b>	<b>109</b>
<b>Chapter 9 Conclusions.....</b>	<b>113</b>
9.1 <b>Findings and contribution to theory and practice.....</b>	<b>113</b>
9.2 <b>Limitations and future work.....</b>	<b>114</b>
9.3 <b>Conclusion .....</b>	<b>116</b>
<b>Chapter 10 Annexes.....</b>	<b>119</b>
10.1 <b>Annex A. Code.....</b>	<b>119</b>
10.1.1 package.json .....	119
10.1.2 truffle.js .....	120
10.1.3 webpack.config.js .....	121
10.1.4 Migrations.sol.....	122
10.1.5 EnergyCommunity.sol .....	122
10.1.6 1_initial_migration.js.....	133
10.1.7 2_deploy_contracts.js.....	133
10.1.8 app.js.....	133
10.1.9 Index.html .....	153
10.1.10 app.css.....	158
10.2 <b>Annex B. Layout of the web app.....</b>	<b>161</b>
<b>Chapter 11 Bibliography.....</b>	<b>171</b>



## *List of Figures*

Figure 1. Centralized vs distributed payment system. ....	12
Figure 2. Hash rate of Bitcoin mining over the years .....	17
Figure 3. Alastria permissioned network types of nodes .....	20
Figure 4. The Blockchain application stack .....	21
Figure 5. Global warming. Temperature change each year .....	27
Figure 6. Renewables energy production in Europe in the last two decades .....	28
Figure 7. Evolution of photovoltaic panel prices.....	29
Figure 8. Californian duck chart.....	30
Figure 9. Evolution of lithium-ion battery prices .....	31
Figure 10. Number of Local Energy Communities in Europe by country.....	33
Figure 11. Energy community governed by a Blockchain platform schema .....	42
Figure 12. Flow of value inside the community.....	46
Figure 13. Bitcoin and Ethereum consumption of electricity in contrast with residential consumption in chosen countries.....	54
Figure 14. Inputs and outputs of OBAMA.....	62
Figure 15. The use of a battery flats the demand curve .....	64
Figure 16. Simple smart contract.....	68
Figure 17. Ganache layout.....	70
Figure 18. Metamask pop up window.....	72
Figure 19. Architecture of the Dapp .....	75
Figure 20. Constructor code .....	78
Figure 21. Function to add users.....	79
Figure 22. Buy EC Coins function.....	80
Figure 23. Truffle compile command run on the community Dapp .....	82
Figure 24. Deploy contract migration code .....	82
Figure 25. Truffle migrate command for the community Dapp.....	83
Figure 26. Information for hour 1 in data.json .....	84
Figure 27. Web App initial screen .....	85
Figure 28. Events tab .....	87
Figure 29. Graphs for measurements of smart meters inside the energy community for user 1 from hours 1 to 24 .....	88
Figure 30. Economic transactions and accumulated balance for the user and DSO .....	89
Figure 31. Accessing the smart contract from the JavaScript file .....	91
Figure 32. Time line of most important events in the platform.....	93

---

Figure 33. Annual demand of the 3 members of the community .....	94
Figure 34. Aggregated demand of the 3 members.....	95
Figure 35. Demand and prices for a week in August.....	96
Figure 36. PV and demand curve in August and in April .....	97
Figure 37. Sample of the output of Obama .....	99
Figure 38. State of charge of the battery .....	99
Figure 39. Cash flows of the totality of the SCP tokens of the Energy Community....	100
Figure 40. Community, PV and Battery smart meters measurements .....	104
Figure 41. DSO economic transactions and accumulated EC coin balance.....	104
Figure 42. User 1 demand and economic flows .....	106
Figure 43. User 2 demand and economic flows .....	107
Figure 44. User 3 demand and economic flows .....	108
Figure 45. Revenues coming from the total amount of SCP tokens each month .....	110
Figure 46. Depreciation of SCP tokens over the useful life of the technologies.....	111

## *List of Tables*

Table 1. State of the art: Theses and papers .....	36
Table 2. Current Blockchain projects for Energy Communities .....	39
Table 3. Dual token system .....	46
Table 4. Terms registered every hour .....	48
Table 5. Configuration of the Energy Community of the prototype .....	57
Table 6. Investment costs for the installation of the technologies.....	59
Table 7. Owner's tab functionalities and information.....	86
Table 8. Functionalities and information shown in the different tabs .....	87
Table 9. Allocation of power contracted among users.....	95
Table 10. Battery and contracted power inputs of OBAMA.....	98
Table 11. CBA analysis calculations .....	101
Table 12. SCP tokens initially bought by each user .....	102
Table 13. EC balances at the end of day 1 .....	105
Table 14. One year electricity bill of the users inside the Energy Community.....	109



## Chapter 1 INTRODUCTION

In the last two decades, and more drastically in the last one, the majority of developed countries are contributing to an energy transition to fight against climate change that is causing environmental catastrophes such as glacier melting, desertification, extinction of species and sea level rise.

This energy transition is driven by the increase of renewable energy in the global power mix, either to cover increasing demand, or to substitute fossil fuel sources that emit greenhouse gasses to the atmosphere that contribute to the global warming.

In order to meet global shared commitments, such as the one set by the European Commission, which states that 20% of energy global consumption should come from renewables by 2020 or as the one fixed in the Paris Treaty that said that CO<sub>2</sub> emissions should be reduced to 40 % by 2030, several subsidies have been given by national governments to incentivize the installation of renewable generation. However, big company suppliers that have a majority of their power mix formed by fossil-fuel sources are more reluctant to change their business models. This, along with the development of information communication technologies and the citizens' growing environmental concern, is contributing to the fact that a growing share of clean energy installations is placed at a distributed level as part of the so-called Distributed Energy Resources (DERs).

Due to the big reduction in prices in Photovoltaic (PV) panels and their relatively small space requirements, PV is the technology most widely adopted by consumers to start producing their own energy. However, as PV panels can only provide energy when the sun is rising, other compatible solutions are needed specially to cope with the high demand peaks that may occur as in Spain in the late afternoon and early morning.

In order to balance the generation and demand at a local level, a first solution would be to store the energy in batteries when there is a surplus of generation for a later use. Batteries have not yet experienced mass adoption due to their still high prices compared to their short useful life what made them not profitable enough. However, batteries prices are being reduced at a fast rate, and they are starting to be considered as a real cost-effective option.

Second solution comes from the popular social trend “*sharing economy*”. Consumers are moving from being product-oriented to service-oriented, they look for experiences, and they have discovered that sharing cannot only benefit themselves in reducing costs or earning benefits from the things they own, but also, it can benefit the ones surrounding them which provides a good feeling. Companies like Wallapop or Airbnb have already

proven that a successful business model can be made out of this trend. Differences among the business models surrounding this concept of sharing economy do exist and while some are non-profit based initiatives other are for profit companies that unlock value from shared assets.

In the case of PV energy, panel owners could trade their surplus of generation to their neighbors and earn more money than what they get when selling it to the central grid (usually because of inefficient or distorted price signals). On the other hand, neighbors could get clean and cheaper energy than the one they get from the central grid. Another possibility, could be to share the ownership of a PV panel among several neighbors, distributing earnings adequately. However, sharing Wh of energy can end in an arduous arrangement of small payments. In cash, it would be administratively practically impossible and by online payments, it would mean paying an intermediary a too high fee compared to the amount of money being transferred.

In this scenario is where Blockchain technology appears to bring a solution. Blockchain, the underlying technology behind Bitcoin and every other cryptocurrency. Blockchain, is a distributed ledger public and visible to all participants that does not require the need of an intermediary to verify transactions but rather it follows the rules written in a programmed protocol to do so. Blockchain allows to cost-efficiently transfer value electronically without the need of a central intermediary. Blockchain main benefits for this use case are that it can automate processes and reduce transaction costs.

This thesis aims to show a possible solution for sharing energy inside a local community with the use of Blockchain. In order to do so in a practical way, a prototype was programmed and tested. The main objective of this thesis is to build an application with the use of Blockchain to learn as much as possible about the technology from its insides as well as to show some value with a technological solution that can help in the energy transition towards a more sustainable world and adapt to consumers preferences. In addition, it was intended to use some important trends that are shaping the society such as renewable energy sources, sharing economy and Blockchain technology.

Chapter 2 is dedicated to Blockchain technology. It explains its origin and its functioning, what the main types of blockchain are and their main consensus protocols. In addition, it is also discussed the industries that Blockchain is called to disrupt. 2.1 revises the main trends that are shaping the evolution of the energy sector and it presents in greater depth the concept of Energy Communities which is the main focus of the project. Once Blockchain and the current electricity industry situation has been explained, Chapter 4 depicts the state of the art of the use of Blockchain for sharing energy between peers.

Chapters 5 to 8 form the second block of the thesis. Through them, the solution ideated and the prototype built are presented. Chapter 5 discusses the characteristics of the

solution as well as the functioning of the platform. In addition, the main hypotheses taken for the prototype created are presented. Chapter 6 shows the optimization tool used for the Energy Community, Chapter 7 explains in detail the Distributed Application (Dapp) that was programmed and finally Chapter 8 shows the results of the simulations done and the main analysis drawn out from them.





## Chapter 2 **BLOCKCHAIN TECHNOLOGY**

### 2.1 **BLOCKCHAIN AND ITS ORIGIN**

---

Simply put, a blockchain can be defined as a distributed ledger. It contains a list of records that are linked one to another and cryptographically secured. All participants of the network share an identical and unique view of the ledger in their computers. There is no central authority in charge of the maintenance and operation of the network, conversely, the participants are the ones that have the responsibility of verifying the new entering transactions using a predefined consensus protocol.

Blockchain allows the guarantee of the uniqueness of a digital asset without the need of a trusted third party. In its place, a system based on economic incentives embedded in the technology encourages a pool of unknown participants to dedicate resources to ensure the non-duplicity of digital assets. This new paradigm permits the execution of transactions and the digitally exchange of value with the assurance of absence of double spending.

Taking as an example an online payment, in the traditional way, the participants put their trust in a bank that is in charge of solving the problem of double spending and basically subtracting the price of the product into the buyer's account and adding it into the seller's one. On the contrary, with a blockchain, the online payment is directly uploaded into the common data base following the rules marked in the protocol. The participants of the network themselves oversee the validity and correctness of the transaction. Figure 1 show a schema of these two types of transactions and it is pointed out who is in charge of keeping the records [1].

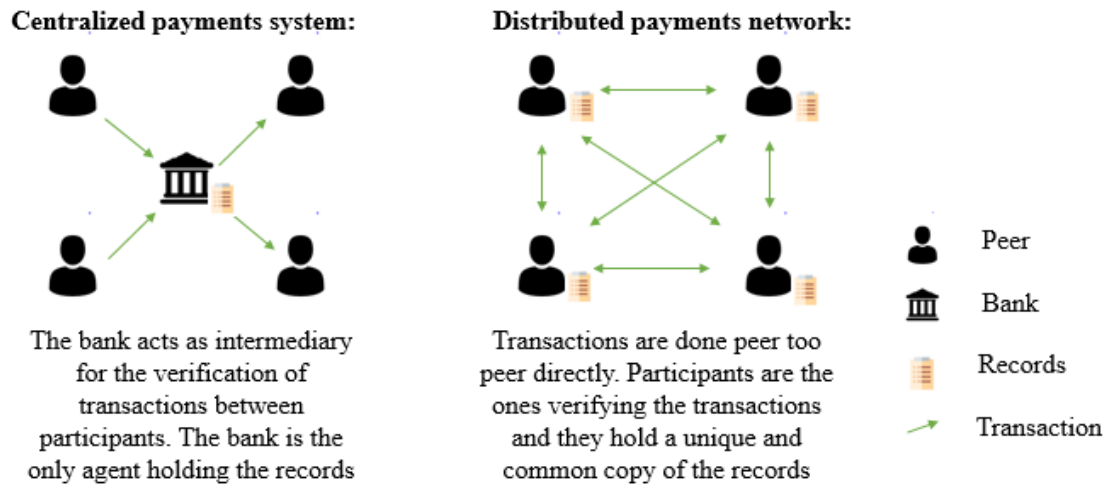


Figure 1. Centralized vs distributed payment system.

Source: Own elaboration

It was not until 2009 with the apparition of Bitcoin that it existed a system able to solve the double spending problem without the need of an intermediary.

On October 31<sup>st</sup> 2008 the white paper “Bitcoin: A Peer-to-Peer Electronic Cash System” was posted to a cryptography mailing list published under the name of Satoshi Nakamoto. It is still not known who this person or group of people are and there is no trace of them. This paper explained the creation of a new decentralized system for making payments and its functioning named Bitcoin. In January 2009, the first transaction in Bitcoin is done and the first block is mined. [2]

However, Bitcoin was not created out of nothing. Prior to it, other digital cash technologies had been invented such as Bit Gold or Hashcash and a lot of work had been done around the consortium of Proof of Work (PoW) and cryptography. Most of this work was developed by a group people that called themselves *Cypherpunks* who were against the capitalist and centralized system and wanted to create something that could make banks obsolete [2].

With time, Bitcoin network spread and it started to be used by more and more people. Thanks to the fact that it is open source, other cryptocurrencies and similar protocols also emerged. The most popular one is Ethereum, created by Vitalik Buterin who extended the scope of blockchain to allow not only value transaction but the distributed processing of any possible algorithm.

## 2.2 EXPLANATION OF BLOCKCHAIN THROUGH ITS MAIN TERMS

---

This section tries to give a deeper insight into Blockchain technology by explaining the main terms of which it is composed.

**Node:** A node is each of the computers connected to the blockchain.

**Transaction:** As explained before, a blockchain is globally shared. All participants have visibility on the entries of the database, but every time a participant of the network wants to change any value inside of it, for example reduce his account by one Bitcoin by transacting it to another account, it has to be accepted by all the other nodes. This change in the state of the network is what is called a transaction. Furthermore, the transaction must be cryptographically signed by the sender [3].

**Block:** When transactions are sent, they are put into a waiting list. Miners will bundle the waiting transactions, following the rules marked in the consensus protocol, into what is called a block. Then they will execute and distribute the new block to all nodes and it will be connected to the last existing one, forming the Blockchain. One may be worried about the possibility of two contradictory transactions being uploaded into the same block, for example if an account has 1 Bitcoin and its owner sends two consecutive transactions of 1 Bitcoin each. But the truth is that when the miner groups the transactions they are put in an ordered list, the one that ends up being in second place will be rejected and will not form part of the final accepted block [2]

**Cryptography:** Cryptography is the art of writing and solving codes to achieve secure communication. In Blockchain, it is more related to making sure who sends a transaction and that the past cannot be tampered with [2]. Cryptography uses mathematical peculiarities to build one-way functions, which are those that transform a piece of data into another one by using an operation that makes it very difficult to get back from the result to the input. An easy example would be the easiness to mix two different colors to get a new one, but the difficulty to get back from the mixture to the original colors. Bitcoin uses the so-called hash function by the algorithm SHA-256.

**Hash:** A hash is a unique identifier of an input of data that has been cryptographically converted by a hash function. In Bitcoin, by using algorithm SHA-256, any input of data will be converted into a hash value with a length of 256 bits. The main features that make hashes so important are: 1) they are unique, it is impossible to find two pieces of data that have the same hash. 2) If any punctuation or letter is changed, the resulting hash is completely different. 3) No matter what the length of the input piece of data is, the hash will always have the same length.

As an example, the hash for the word “Bitcoin” is:

b4056df6691f8dc72e56302ddad345d65fead3ead9299609a826e2344eb63aa4

while if the capital letter B is changed to a lower case b. “bitcoin” the resulting hash is:

6b88c087247aa2f07ee1c5956b8e1a9f4c7f892a70e324f1bb3d161e05ca107b

which is completely different than the one before. In addition, the hash for the whole content of Bitcoin’s Wikipedia is:

da34df06501a1f148efc9937107659a7f64611390e2215eadebb889697b7a83b

which is the same length as the one for the word “Bitcoin”.

**Public and private keys:** When becoming a node of the network, the participant is assigned a private key and a corresponding public key. The public key is the identifier of the account. Every time a transaction is made, the public key of the sender and of the receiver are recorded and every node is able to see that a transaction occurred between those two accounts. The private key is only known by the holder of the account and it serves to sign and authorize transactions. This pair of keys has two applications: 1) The public key can be used to encrypt something, and the private key can decrypt it. This is how secret messages can be sent, albeit in one direction. 2) The private key is used to encrypt something and the public key is used to decrypt it proving that it could only have been encrypted by its corresponding private key, although it is still kept unknown [2].

**Mining:** it is the process of verifying the correctness of performed transactions and inclusion in a new block. There are some nodes that take part of the consensus building that are called miners. In the Bitcoin Blockchain it is used Proof of Work (PoW) consensus for mining. PoW can be defined as a competition to solve a difficult problem. Miners group some transactions and hash them, afterwards they apply a random number to them called *nonce* to change the hash so that it fulfills the problem rules. In Bitcoin the mining rule is that the resulting hash has to start with a specified number of zeros. This number of needed zeros is called the *Mining Difficulty*. As for May 2018, it is set to 18 zeros. In order to get the correct nonce that transforms the hash into one that begins with the specified number of zeros, a proof and error system is used. Explained in other words, each of the miners apply a number to their transactions hash, let’s say a 0, they hash everything again, if the resulting hash starts with 18 zeros they stop and they will have found next block, if not, they try with another number, for example 1 and repeat the process until they find the first nonce that serves them. The miner that gets the result the first is the one that has the privilege to create the new block and gets an incentive out of it.

**Consensus Protocol:** It refers to the rules set for the agreement between nodes for creating next blocks and therefore for the correct operation of the network. Section 2.3 discusses it in more depth and describes the two most important types of protocols.

**Gas:** As it can have already been inferred, the process of mining is vital for the working of the network and the continuous addition of blocks to the chain. In order to keep miners active and continuously solving hard computation problems, that cost them money in terms of electricity and powerful GPUs (Graphic Processing Unit), it is necessary to give them an incentive. This incentive is formed of two parts: 1) New Bitcoin are created and given to the winner miner, 2) each transaction has a fee so that it is included in the block by the miner. The winner will also acquire all the Bitcoin resulting from the transaction fees. In Ethereum this fee is called gas. The higher the gas payed by the sender of a transaction, the highest incentive for the miner to include that transaction into the block. If a very low gas is payed, especially when the network is more congested, it is probable that the transaction stays in the waiting list for a very long time. If this happens, at a certain point the transaction is refused and it is asked to the sender to use a higher gas. In addition, a fixed floor limit of gas is specified for each transaction depending on the computation is needed to run it.

**Smart Contract:** A smart contract is a piece computing code that is processed by decentralized resources when triggered by some conditions. When executed it generates results that are stored into the blockchain and therefore are made tampered proof. The strength of smart contracts is that they give assurance that the terms written in them are going to be strictly followed and no trust is needed to be put into a third party.

**Oracle:** Oracles are gates to the outside world of blockchains, to the *real world*. They feed smart contracts with information from outside the chain. They can appear in the form of sensors, IoT devices or web pages [2].

**Cryptocurrency:** It can be defined as decentralized digital money that is used to fuel decentralized value transfer systems. Cryptocurrencies are an inherent part of the Blockchain as they are a fundamental element of the consensus mechanism that incentivizes participants to defend the integrity and truthfulness of the transactions. Each blockchain has its own cryptocurrency used to transfer value between accounts. The most popular ones are Bitcoin and Ether.

## 2.3 TYPES OF CONSENSUS PROTOCOLS

---

---

As already mentioned, the consensus protocol is key for the functioning of Blockchain. It is the group of rules and economic incentives that define how transactions must be validated and guarantees (or at least fosters) the constructive behavior of participant agents/nodes. Consensus means that nodes agree on the same world state. This is a very difficult thing to achieve in any network without a trusted and centralized authority because it is difficult to ensure that all the participant computers will defend the truthfulness of the same reality.

A simple way to understand a consensus mechanism is to think of it as a set of incentives that forces participant nodes to commit resources, efforts or value to defend the truth of a certain reality based on a set of predefined rules.

A very used metaphor to explain this is the so called **Byzantine Generals' Problem** that states the following:

There are two allied generals that have to attack the enemy at the same time or they will fail. In order to get to an agreement to decide at what time they are going to do it, they can send messages to each other, but they will never be 100% sure that the message reached the other general. General A can send a message to general B saying "*the attack will be at 8 a.m.*", but he cannot make sure that the message reached its destination. If general B sends back a message saying that he received it correctly, he will not know if general A got the acknowledgement. And this can go on indeterminately. The conclusion is that it is uncertain that messages arrive to their destination when the network is unreliable, and peer-to-peer Internet connection is unreliable. It is even more difficult to make sure that messages are reaching its destination if instead of having two points of connection, there are thousands of them like in a decentralized network.

Bitcoin solves this problem with the **Proof of Work (PoW)** consensus. Instead of having a negotiation process to reach an agreement, this method punishes unwise decisions: if a node guesses what the world state is and later turns out to be unaligned with the rest of the nodes, then all its mining work will have been done in vain. Therefore, all nodes make their effort to identify what all other nodes will estimate about the world state. This way they will be synchronized.

To avoid a situation in which all nodes propose solutions and no one listens, a proposal is only accepted when the node solves a hard-to-solve puzzle. Once the first node solves the puzzle, explained in previous section, it will send to the rest of the nodes its proposal that is the new block containing a group of transactions, and they will verify them. This way is how synchronization is reached by all nodes in a PoW blockchain.

The total gear employed across the network to solve these puzzles determine the **hash rate** of the network which is related to the Mining Difficulty already mentioned. This hash rate is refreshed every 2,016 blocks and it changes depending on the number of miners. The more miners there are trying to solve the puzzle, the more difficulty is applied to it, the higher the hash rate gets and therefore the more secure the network is.

Figure 2 shows the hash rate evolution of Bitcoin mining since its origins in Tera hashes per second. In 2018, for finding the correct nonce that solves the puzzle, over 35,000,000 Tera hashes per second are tried each second across the network to find the correct nonce.

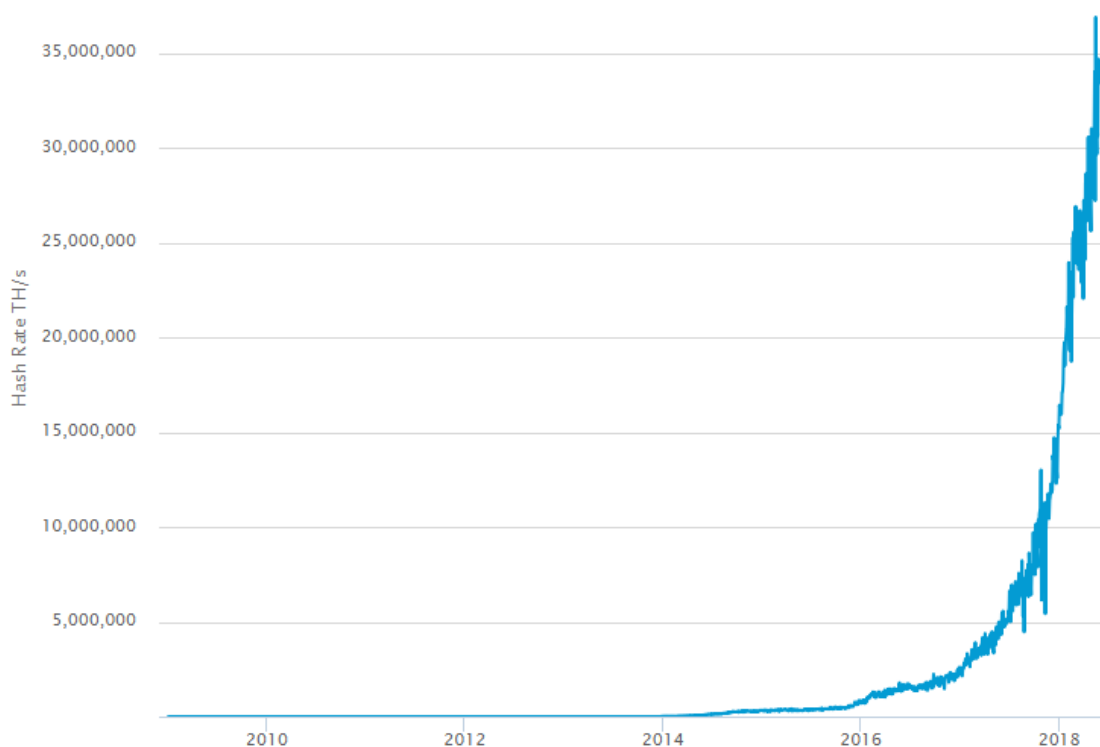


Figure 2. Hash rate of Bitcoin mining over the years

Source: [4]

It can be seen that in the last two years, the hash rate has wildly increase. This has occurred because the price of Bitcoin has become very high and mining has been converted into a lucrative business with a lot of new entrants.

Currently, as for May 2018, the incentive for finding the new block gives the miner 12.5 BTC which is around 95,000 USD plus the transaction fees included in the block. There are 21 million Bitcoin to be mined in total, from which around 80 % has already been mined. Each 210,000 blocks, around 4 years at a rate of one block each 10 minutes, the mining reward is halved. For maintaining the network running and miners being

incentivized, it is believed that while the mining award goes down the fee transactions will go up.

However, looking at Figure 2, it can be inferred that such difficult puzzle to be solved to maintain the network secure starts to get nonsense due to the great amount of electricity that is needed to create all that amount of hashes per second. As it is deeper discussed in section 5.3, Bitcoin spends annually more electricity than Iceland.

This is why other types of consensus are being proposed. The most popular one that a big number of projects are already using is called **Proof of Stake** (PoS).

In this consensus, instead of rewarding the winner of a race to solve a difficult puzzle, it is punished the miner that tries to split the network, that goes for a state of non-consensus. PoS is based on a mechanism of bonds. The chain holds a bond, or stake, for every miner, randomly, in a deterministic way, next miner to create the new block is chosen, if he does something that goes against the protocol, his stake is retrieved. There is no reward for mining the new block apart from the fee transactions [2].

This consensus, although it does not provide the same security against attacks as the PoW it is much more efficient. Consensus protocols and mechanisms are a pivotal component of blockchains and as such, have become an exciting area of research for mathematicians, computer scientist and economists.

## 2.4 TYPES OF BLOCKCHAINS

---

Regarding accessibility options, Blockchains can be named Public, Permissioned or Private.

A **Public Blockchain** is a fully decentralized network where there is no restrictions or barriers to new entrants. Anyone can look at the transactions happening in a public blockchain and can start a new node to make transactions anytime he wants. Bitcoin and Ethereum are public blockchains. They are the purest and original form of blockchain, as Satoshi Nakamoto ideated it.

Due to its nature of permission less, it is more exposed to possible attacks and needs of consensus as PoW, already explained, to ensure the security of the network.

On the other hand, a **Private Blockchain** allows only the entry to selected and verified participants. Unlike public blockchains, in a private one there is a network operator that invites and validates the incoming participants, it is in charge of maintaining the network



and decides the rights of mining and rewards. The operator can delete or override the entries in the network when required. [5]

For this type of blockchain there is no need to create complex consensus and therefore mining can be kept fast and efficient.

Private Blockchains are interesting for corporations that want to automatize their processes and make them more efficient, to have control of the network and apply the fees they want.

A **Permissioned Blockchain** is a mix of the two above. They can differ a lot between them and can be customized very differently. They can allow anyone to join the network after suitable verification of their identity, as well as they can grant special permissions to each participant so that they are restricted to their functions. So for example, a participant can only have the right to read but not to write, or there can be validator nodes which sole function is to validate blocks. [5]

It is worthy to mention the permissioned Blockchain being developed in Spain called **Alastria** that comprises more than 150 national institutions and that is applying the needed rules so that the ecosystem complies with the European legislation. Its aim is to provide a standard methodology for the exchange of information across different sectors by means of smart contracts. [6]

Figure 3 shows the different types of nodes that Alastria has. As explained before, participants have permission to specific roles. Not all nodes are equal and have access to all functions inside the network. Client-nodes in Alastria have to be validated and have to pay a subscription fee to be able to operate in the network, similarly to what occurs in a Private Blockchain. However, distinctly to it, validation of blocks is distributed and the system operator cannot do and undo as he wants.

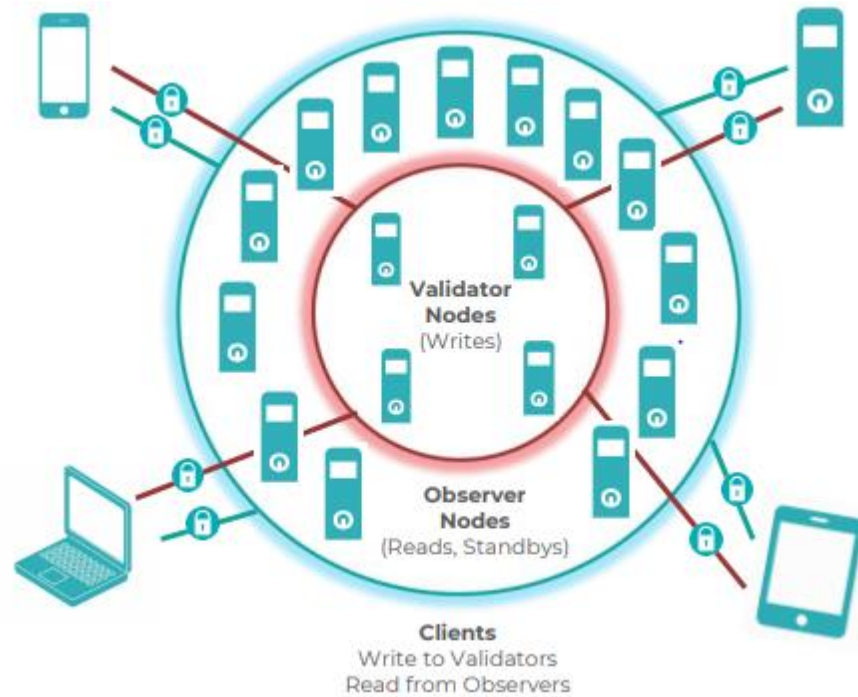


Figure 3. Alastria permissioned network types of nodes

Source: [6]

## 2.5 BLOCKCHAIN STACK

This section explains the architecture of applications that are built on top of a Blockchain. Figure 4 shows this architecture also called stack.

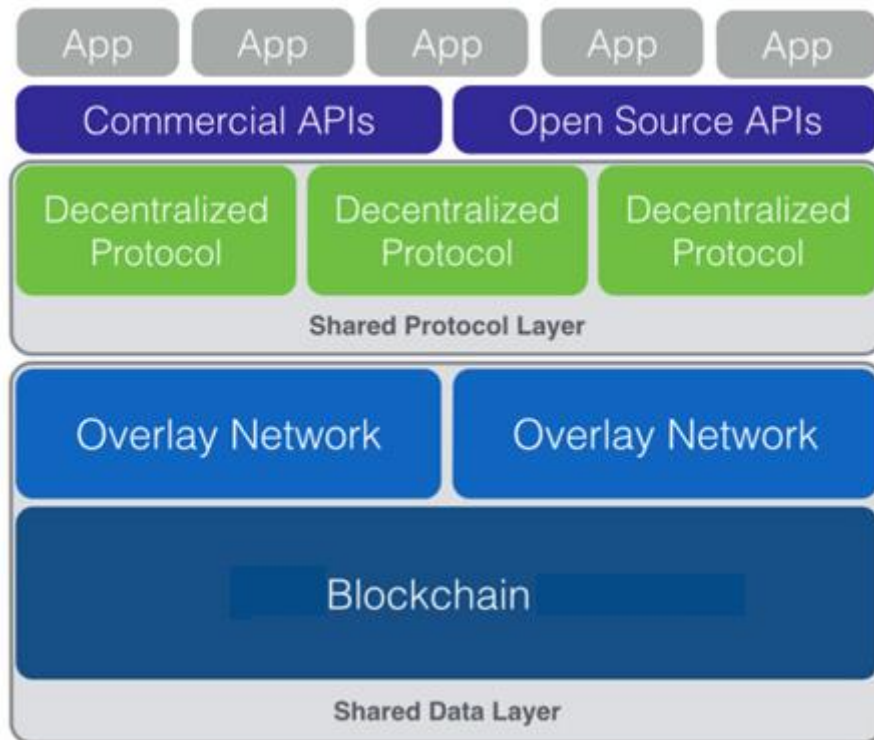


Figure 4. The Blockchain application stack

Source: [7]

Everything inside the two gray rectangles, from now on, shared Data and shared protocol layers, is decentralized. Nobody controls these parts of the system and they are accessible to anyone. It can be seen that the higher in the stack, the thinner the layers are. It is also noticeable that the decentralized part occupy more less 80% of the full stack. Traditional internet applications are built in top of decentralized technologies as HTTP, but they only form around 15% of the full stack, being the majority of it centralized and private. [7]

Going from bottom to top, the shared data layer is form by the Blockchain, so for example Bitcoin Blockchain, and overlay networks. An overlay network is a network that is built parallel to the Blockchain to be able to perform tasks that are not available in it but that still makes use of it for validating the transactions.

On top of the Shared Data Layer, the protocol can be found. This layer is explained with detail in section 2.3.

Since it is difficult for an average programmer to build on top of protocols, APIs are created exactly for that. They are an intermediate layer that makes it quicker and easier for programmers to build their applications on top of decentralized protocols [7].

Finally, the applications are the consumer facing part of the stack. For users, they will be working very much alike conventional apps. Applications that are built on top of a decentralized network are called Dapps (Distributed Apps) and the main difference with conventional apps is that they are able to communicate to each other since they are sharing network and protocol.

## 2.6 SECTORS TO BE DISRUPTED BY BLOCKCHAIN

---

Blockchain is called to be a technology that disrupts numerous industries because of attractive characteristics that are explained below.

First, it brings more **security** to the system. Due to its distributed nature and the applied protocol, information and value stored are more secured than in a centralized data base.

Second, it brings **transparency**. Members of the blockchain can view all transactions and can make sure that what written there is not going to be changed or duplicated. There is no need any more to trust on a third party. In addition, due to the elimination of the intermediary, costs are reduced.

Third, due to the immutability and non-duplicity that the network provides, Physical assets can be digitally or virtually represented for different purposes such as proof of ownership or traceability. This process is called **tokenization**. It can also be useful to divide the ownership of an asset into different individuals. It can be seen similar to the use of shares. The good thing of tokenization is that tradability of these assets can be made faster and more efficient without the use of a notary, and thanks to the use of smart contracts, these changes can be subject to some condition. This opens a new range of possibilities that were not efficient in money and legal bureaucracy before with the use of notaries. As an example, imagine that an apartment to be rented is tokenized. Investors buy tokens of the apartment and each month they earn money out of the renting depending on the number of tokens they have. Furthermore, any time they want, they can sell their tokens to other nodes in the network. This new system brings efficiency and liquidity to investors and markets.

Fourth, the possibility of transferring value peer to peer without intermediaries opens the possibility of making **micropayments** with no commissions. The complex process of securely transferring value digitally, is embedded at the core of the blockchain technological approach.

And fifth, thanks to smart contracts, **processes can be automated**, saving costs and simplifying systems.

Due to all these qualities explained, Blockchain can have many applications inside different sectors that can change a lot the way of doing things, making processes more efficient and economic. Below, it is depicted some examples of application in industries where Blockchain could have a big impact. They are mostly taken from reference [8].

#### Banking and finance:

Industry in which Blockchain has hit first and where it is having the biggest impact. The new internet of value allows no commission peer to-peer transactions of money regardless of the country they live in and the bank they are affiliated to. *Circle*<sup>1</sup> is a start-up that permits users to send money with no commissions.

#### Insurance:

Five of the most important insurance companies: Aegon, Allianz, Munich Re, Swiss Re and Zurich, have already grouped and formed a Blockchain consortium called “Blockchain Insurance Industry initiative-B31” to study the different applications that this new technology could have in their field.

For example, Blockchain could enable car micro-insurances. A driver could send information of its record of accidents, in an anonymous way, to different insurance companies for insuring his car for a trip he is going to do for one week. The insurance company depending on his record, which has been previously verified and is immutable, can offer him different prices for his trip.

#### Supply-Chain:

With Blockchain, traceability can become easier to achieve. Sectors such as pharma or luxury brands lose great amount of money with the falsification of their products. In addition, in pharma or food industry, is not only about money, but also about the health of the consumer that is being put under risk. With a Blockchain supply chain solution and the use of sensors, products can be traced from their origin or manufacturing to the store in which it is sold. Designation of origin can be proved 100% to customers.

Similarly, sectors like art and music can be benefitted from Blockchains by enabling the proof of originality of the piece and its author.

#### Energy:

---

<sup>1</sup> <https://www.circle.com/en-gb/>

In the energy sector, which is the main focus of this document, Blockchain can mainly help in the development of distributed renewable generation and the introduction of the electric vehicles.

With the increasing apparition of *prosumers* (i.e. consumers that also generate electricity) the number of agents that want to form part of the market increases. Peer to peer sell of electricity can be facilitated by the use of Blockchain.

Blockchain can decentralize the charging of electric vehicles. Users can enable their electricity source to other users so that they can charge their car and pay the owner of the plug directly via an app connected to the Blockchain. *Share&Charge*<sup>2</sup> is already working on implementing this in Germany.

---

<sup>2</sup> <https://shareandcharge.com>

## Chapter 3      **EVOLUTION OF THE POWER INDUSTRY**

### **3.1 STRUCTURE OF THE ELECTRICITY GRID AND MARKET**

---

The electricity grid can be divided into four well distinguished blocks: generation, transmission, distribution and retail.

Power plants generate electricity out of different resources, then it is transformed from low to high voltage for its transport. Transmission lines are characterized for traveling long distances to bring electricity to different towns, then the electricity is again transformed to low voltage for its distribution to the consumers. High voltage is used in transmission because it is more efficient as it generates less losses which is important when traveling long distances. On the other hand, distribution lines use low voltage for increasing safety where inhabitants live. Retailers are in charge of commercializing the electricity to the consumers.

Transmission and distribution are natural monopolies completely regulated, while generation and retail are open to competition to private companies. However, this was not always like that, before the 1990s, the sector was fully centralized and regulated under the so called traditional model. There were vertical integrated companies that did all four actions: generation, transmission, distribution and commercialization.

During the decade of the 90s, most of European countries started the transition from the traditional model to the so-called market model. Generation and retail changed to competition mechanisms with a process of deregulation and the subsequent privatization of state-owned companies and opening the market to new entrants. The government does not fix the electricity price anymore but tries to ensure perfect competition. The transmission system (TSO) and distributed system operators (DSO) both still operate under strict regulations due to their characteristics of natural monopolies.

TSOs are in charge of operating and maintaining the grid, making electricity available to consumers, extending the network when necessary, balancing power demand and supply and controlling voltage frequency. For all this they charge a tariff to consumers.

The main factors for this change were:

- Reduce high prices imposed by the centralized monopoly that were making the consumers unhappy. There were also unjustified price differences among areas.

- Reduce intervention of the State plus a need to reduce public debt (need of new investments by the private sector).
- Continuous increase of the interconnected system that needed of a more efficient functioning.
- Increasing introduction of generation technologies that point towards reducing the size of generators and increase the number of potential investors.
- Development of measurement and telecommunications systems that allow the entrance of new and diversified agents.

With the new restructuring of the system, there is free access to the market in both sides, supply and demand. There are two markets: wholesale and retail. In the **wholesale market**, generators sell electricity to retailers, and in the **retail market** the retailers sell the electricity to their customers. Consumers can choose the retailer they want and any retailer or generator can start participating in the wholesale market when having the appropriate license.

The wholesale or pool market can be executed by two possible procedures: Periodic clearing process and continuous auction. In the **periodic clearing process**, generators make bids for the prices at what they are willing to sell their electricity for the next 24 hours. For each hour where the offer and demand curves intersect, the clearing price is set for all the energy sold at that hour. In the **continuous auction or trade**, the generators are paid the amount they bid, if they have a buyer willing to accept the bidding price. In continuous trading, the scheduling can be seen decentralized since the prices are fixed continuously and not every 24 hours.

After the spot market, the TSO is in charge of running the balancing market. Usually, generators are obliged to keep some reserves in case there is a need of more supply.

## 3.2 TRENDS

---

In this subchapter it is explained the trends that are shaping the market and that are showing the direction towards which the evolution of the sector is moving to.

### 3.2.1 CLIMATE CHANGE

---

In the last century, the Earth is experiencing an increase of the average temperature that has been named as global warming or climate change. Figure 5 shows the change in



temperature that the Earth suffers each year. It can be seen how in the last 50 years it has been increasing every year, and with a trend of each year more than the previous one. The figure shows data measured by different institutions for contrasting purposes.

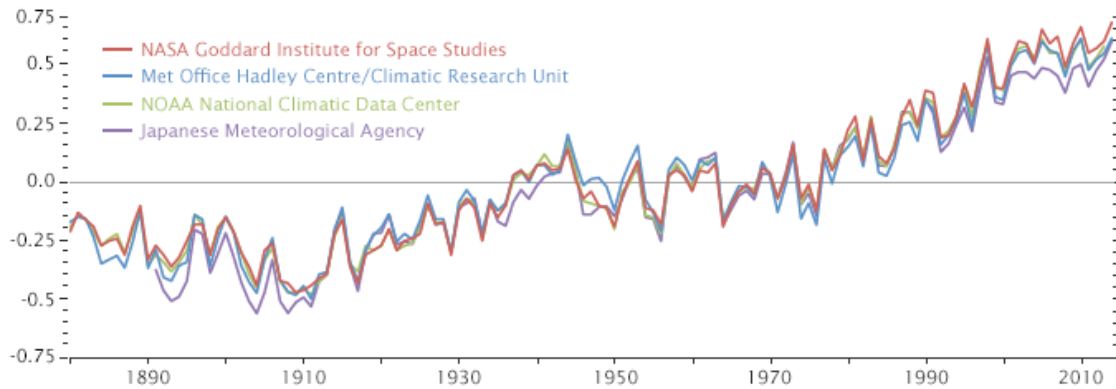


Figure 5. Global warming. Temperature change each year

Source: [9]

Global warming is being caused in big part due to the human emission of a group of gases that are retained in the atmosphere and allow sun light to go through. This makes the global humidity go down and have severe consequences as glacier melting, desertification, extinction of species, sea level rise etc. These emissions are called Green-House Gas (GHG) emissions and are mainly formed by CO<sub>2</sub>.

Because fossil fuel sources are big generators of GHG emissions, their substitution by renewable generators together with the increase of consumption efficiency has become a big concern of importance at a political level in the last decades.

In 1997 in the Kyoto conference, it was globally agreed for the first time at a political level that countries should reduce their GHG emissions studying each of the regions one by one and differentiating between industrialized and in development. Later, at the COP21 in Paris, most of the countries committed to reduce their GHG emissions with the goal of not letting the temperature of the Earth increase by 2°C.

Due to this spread concern, adoption of renewable energy sources has been increasing in the last years. In Figure 6 below, it can be seen the increase in production of renewable energy in Europe (without counting hydro) in the las two decades.

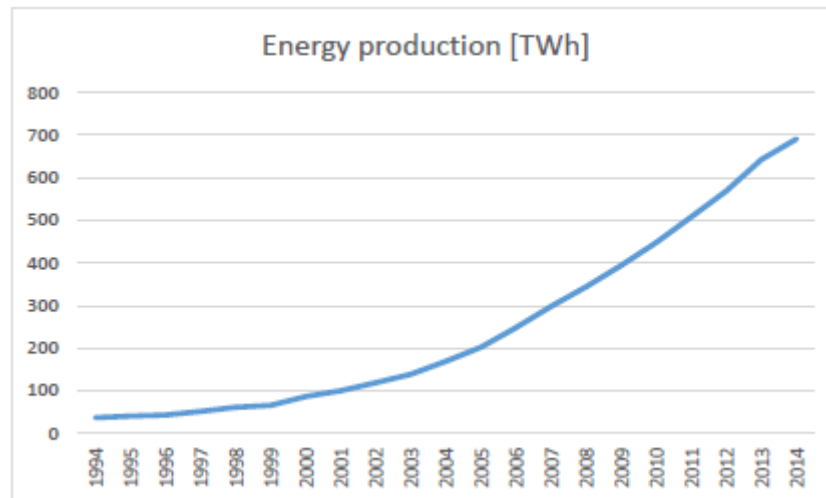


Figure 6. Renewables energy production in Europe in the last two decades

Source: [10]

Furthermore, in January 2018, the European Parliament endorsed the Industry Parliament Commission proposal that asked for the integration of 35% of renewable global energy production by 2030 [11].

### 3.2.2 DECENTRALIZATION

The technologies that are being more used for this power system transition to the inclusion of renewables are being PV panels and wind turbines due to their lately technological development, governmental subsidies and reduction in prices. However, a big portion of this new generation capacity is not being installed as traditionally (big power plants far from towns) but in distribution system with small wind turbines or solar panels farms or by active consumers. Therefore, supply of energy is getting more decentralized.

The cost of wind and solar PV has decreased by 40 % and 60% respectively between 2008 and 2014 [12]. In addition, in Figure 7 it is shown the evolution of the prices of PV panels in the last years and the expected prices in the years to come.

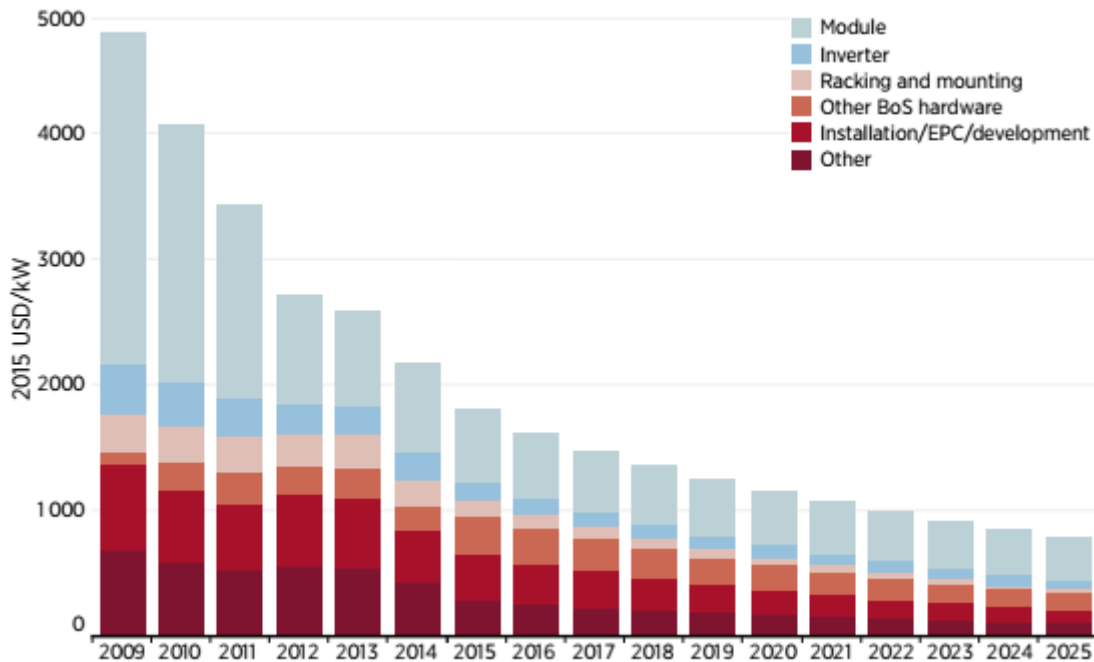


Figure 7. Evolution of photovoltaic panel prices

Source: [13]

Due to the big decrease in prices in PV panels it has become more available for individual consumers to install their own power supplies. In addition, there is a trend that individuals also want to become part of the energy transition and have a higher control of their electricity bill. They are changing from passive to **active consumers**. As an example, In the PJM market in the eastern United States, nearly 1.5 GW of capacity daily comes from the demand side [12].

In addition, according to a study by CE Delft, by 2050, about 187 around 83% EU households could contribute to renewable energy production, demand response and/or energy storage [14].

One of the main problems of renewable energy is that it depends on variable weather conditions. That brings a big uncertainty to the system and can produce imbalances. Furthermore, PV panels only produce energy when the sun is rising. If there is a big penetration of the technology, the net load curve that other energy sources have to fulfill will become low at midday and high at late afternoon and morning. This phenomenon is shown in Figure 8, where for a day of March in California, with a big implementation of PV generation the net load curve would look as a “duck”. This would bring problems of possible over-generation at midday and very demanding ramp needs in the late afternoons.

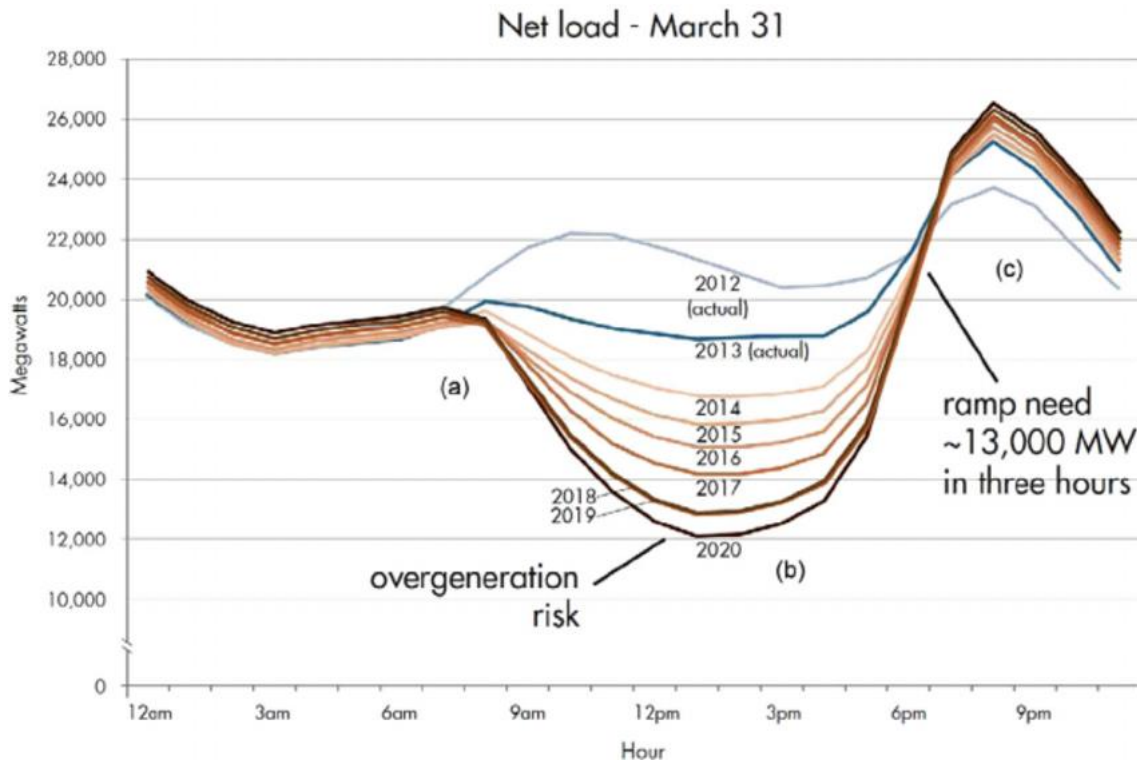


Figure 8. Californian duck chart

Source: [15]

In order to prevent these big differences of net loads at different times of the day and high injections of electricity in the distribution network because of the high peak of PV production, energy storage is becoming important and popular.

Although, batteries are still not very profitable due to their short useful life and high prices, there is being a lot of development that is leading to an important decrease in prices that show that in not long time they will become cost-effective. Figure 9 shows the evolution of Lithium-Ion batteries for the past and following years.

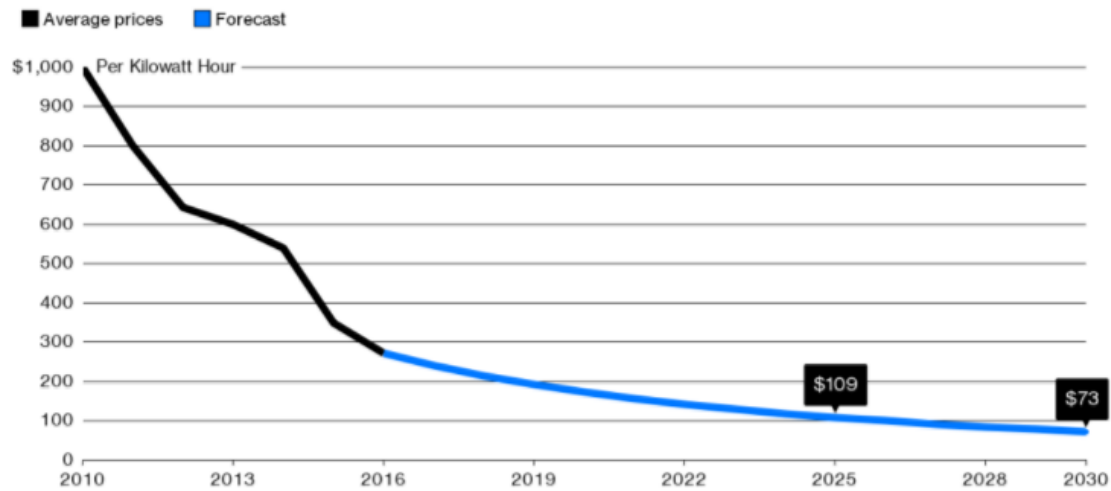


Figure 9. Evolution of lithium-ion battery prices

Source: [16]

### 3.2.3 DIGITALIZATION

In this last two decades, it has also been experienced a proliferation of information communication technologies (ICTs) that enable energy to be produced, transmitted and consumed more intelligently and efficiently. Thanks partly to the increase of ICTs, demand can become more participative in electricity services. Furthermore, there has been a big increase of smart meters implementation, it is expected that for 2020, 72% of European and 100% of Spanish consumers will have one [12].

The integration of generation at a distributed level makes it important for networks to technologically evolve, for instance they will have to be able to transmit flows in both ways. Furthermore, regulators will have to incentivize distribution companies to develop innovative network management approaches. Distribution companies will have to become true “system operators” in the sense that they will acquire network services such as forwarding commitment of DERs capacity to stand by to meet voltage control or congestion management requirements [12].

Digitalization also brings the integration of new capabilities such as the Internet of Things (IoT) that will make possible the digital interconnection of the loads for an intelligent response of demand. With IoT, it will also increase the ability to determine, with increasing granularity in time and space, the economic value of electricity services and appropriate charges for the use of electricity networks [12].

In addition, with the increase of internet-connected devices, cyber security becomes much more important and challenging.

### 3.3 LOCAL ENERGY COMMUNITIES

---

---

Due to the trends mentioned above the congregation of consumers in what is called local energy communities is becoming more popular. With the integration of DERs, the possibility of sharing energy between neighbors becomes more relevant. In fact, with the current trend of sharing economy that has brought a lot of success to projects in other industries such as “Airbnb”, “BlaBla car” or “Glovo”, makes it more reasonable that this kind of congregation might become more and more popular in the years to come.

Following the definition given by the CEER white paper [17], a local energy community is a cooperative/partnership/non-profit organization of final consumers that cooperate in the generation, storage and aggregation of renewable energy or in the improvement of energy efficiency.

By sharing self-generation technologies, consumers can benefit economically since they do not need to make such big initial investments. In addition, they can also achieve social and environmental benefits in the sense that they feel part of a community that fights against the climate change and they can form part of the decision making of local energy politics [18].

Local energy communities have the following values:

- Open and free membership
- Democratic control of the members
- Education and information to the members so that they can participate in the decision making of the community.

Energy communities started to be implemented in Europe in the 1970s in Germany, Denmark, Netherlands and Belgium. However, due to the grid structure of the moment, lack of subsidies and support and expensive renewable sources, it did not have an important follow-through.

It has been in the last years that this form of electricity service is getting more relevant and more active back up from governmental institutions. In November 2016, in the “Winter Package”, the European Commission outlined the important role that Energy Communities may play in the energy transition and incited the member countries to give facilities to consumers to be able to become active consumers and participate in distributed generation of renewable energy. [19]

In Spain despite the instauration of the Royal Decree 900/2015 that included several obstacles for the self-generation and made it more costly, in the last years there has been

some progress towards reverting the situation. In July 2017, the Constitutional Court has eliminated the law that prohibited shared self-generation [20]. Moreover, the retailer Hola Luz has installed the first shared self-generation facility in one household in Catalonia. [21]

Currently, there are 2,397 renewable energy communities in Europe being most of them located in the Occident. In Figure 10 it can be seen the number of energy communities each country has. Germany, Denmark and Austria lead the list with a big difference. In Spain, there are only 20. ResCoops is the European federation of energy Cooperatives and more than 50% of the total energy communities in Europe are affiliated to it. [22]

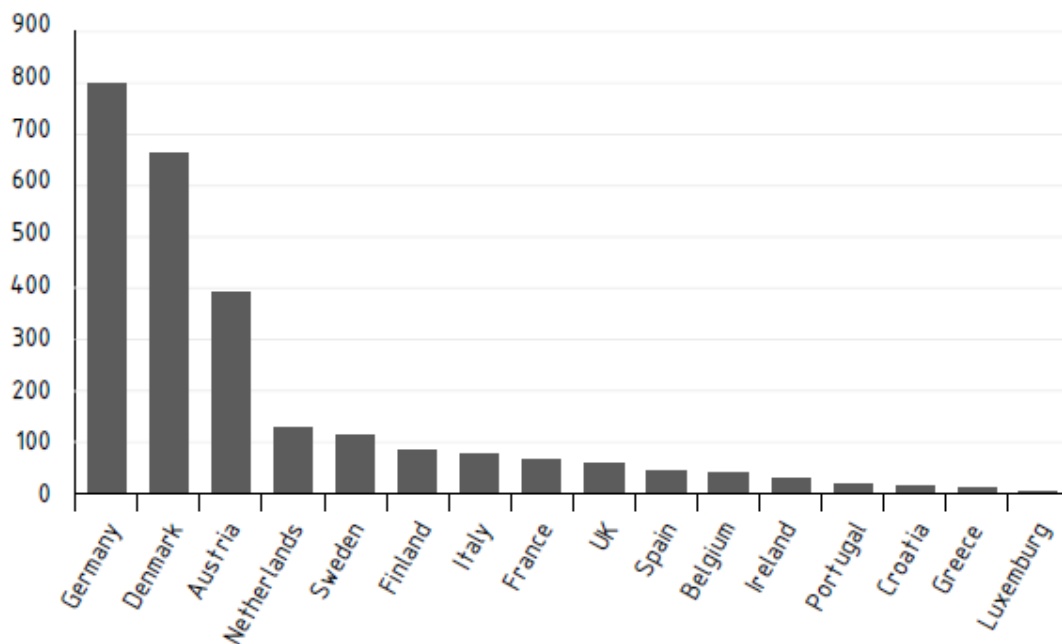


Figure 10. Number of Local Energy Communities in Europe by country

Source: [22]

To sum up this chapter, there is an increase introduction of renewable energy installation at a distributed level, consumers want to take part of the electricity market and the energy transition towards fighting the climate change. In addition, there is being a big penetration of ICTs and smart meters making everything more intelligently connected.

Energy cooperatives might be a suitable way for consumers to allow them to start producing renewable energy due to the advantages of shared self-generation. Furthermore, these communities will need a high level of transparency to show their members that their shares in renewables are really being reflected in their electricity bills. Blockchain technology looks like a suitable tool that could increase this transparency and

give the needed trust to members. It could additionally automate processes and make the accountability of energy and payments easier.

Next chapter will revise the projects that are working on this idea and in Chapter 5 it is presented the solution developed in this thesis.



## Chapter 4 STATE OF THE ART: BLOCKCHAIN FOR P2P ENERGY TRADING AND ENERGY COMMUNITIES

This chapter describes the revision of the literature on the topic of energy communities and more specifically on the application of Blockchain for the exchange of energy between neighbors and local communities.

Theses and scientific articles were studied covering this topic as well as business cases that are been developed in form of start-ups or ICOs proposing similar solutions as the one covered in this document.

Table 1 shows scientific researches done in the topic covered in this thesis.

Name; author	Year	Type of document	Description
Blockchain in the electricity market: Identification and Analysis of business models; Alyssa Orlov; [1]	2017 Autumn	Master thesis	Deep analysis on the business models of all current projects working with Blockchain in the electricity sector.
Blockchain in electricity: a Critical review of progress to date; Eurelectric; [23]	2018, April	Scientific report, white paper	Analysis of the current integration of Blockchain in the different parts of the value chain of the electricity sector, focusing on its potential and its limitations and risks.
Blockchain, an opportunity for energy producers and consumers?; PWC [24]	2016	Scientific report	Analyzes the main roadblocks of Blockchain in the energy sector and the opportunities it can bring to consumers
Barreras y oportunidades para el desarrollo de comunidades energéticas sostenibles en España. estudio comparativo con Estados Unidos y Alemania; María del Carmen Romero [25]	2015, November	PhD thesis	Deep analysis of energy communities and their opportunities in different economic and legal frameworks.
Blockchain for peer-to-peer energy exchanges: design and recommendations; David Vangulick et al [26]	2017	Scientific article	They elaborate a Blockchain solution for an energy community based on criteria such as acceptance of the wholesale/retail market, the resilience of the consensus to approve a block, the accuracy, traceability, privacy and security of the proposed schemes

Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids; Claudia Pop et al [27]	2018 January	Scientific article	It proposes a solution in which a Blockchain platform is used to match demand and generation of a smart micro grid with the appropriate financial settlement for flexibility providers.
Integrated Community energy Systems; Binod Prasad Koirala [28]	2017	PhD thesis	It develops Energy Community models for their optimal planning and operation and assesses their performance based on economic and environmental metrics.
Towards the design of flexibility management in smart grids: A techno-institutional perspective; Cherrelle Eid [29]	2017	PhD thesis	It discusses the flexibility management in different scenarios (centralized, decentralized, micro-grids) and in different levels of the electricity grid structure.

*Table 1. State of the art: Theses and papers*

In Table 1 some of the papers, mostly the theses presented, talk about the use of energy community and their characteristics. They were used to study in more depth about this kind of formation and understand the features and opportunities that they bring to consumers.

Some other papers discuss the integration of Blockchain into the energy sector and what are the main advantages and limitations it has at the different levels of the value chain. During last year, a big number of documents of this sort are being released. They were useful to analyze the pros and cons of using Blockchain technology as it is explained throughout this document and to understand that the Blockchain application that is being mostly used and that has an earlier adoption is the P2P exchange of energy or local community management.

A recent survey found out that 57 % of money raised for blockchain-in-electricity projects is coming from the ones that use blockchain to verify and execute P2P transactions more rapidly [23].

One of the main reasons to develop into a local solution compatible with the current electricity system functioning is because of the structure of the power industry. As explained before, the transmission and distribution network are regulated natural monopolies where a single entity, either a TSO or DSO, provides the service. This means, that these institutions are responsible for multiple functions inside the network such as balancing or security. Therefore, any electricity trading or operation done at a distribution level, even including local P2P trading, must be reconciled with the DSO. Therefore it is difficult to cut intermediaries, which is one of the most important features Blockchain can provide, since these institutions, are needed, at least in the present and in the near future [23].

The solution exposed in this thesis aims to reduce the need of the intervention of the figure of the DSO or TSO as much as possible, so that to make the application as feasible and early adopted as it can get.

Other entries in Table 1 show different applications of Blockchain for P2P energy trading done for scientific research purposes. They served as guide for the elaboration of the solution presented in this thesis from Chapter 5 on. However, to have a deeper understanding on the topic and to see what applications are being deployed in real life, a deeper search was done into startups and business projects that are developing Blockchain solutions for the exchange of energy at a local basis as shown in Table 2.

#	Project; Location	Proposition	Blockchain; Consensus	Phase	Business model
1	Pylon Network; Spain [22]	Through the intelligent smart meter they have created: Metron, they allow prosumers to exchange energy between them. They also want to include cooperatives and retailers in their network.	Private blockchain overlay network of Ethereum; Proof of Collaboration	ICO launched, 0.24 M USD raised- Currently building up the platform	Not defined yet
2	Power Ledger; Australia [30]	They allow prosumers to sell their surplus to their neighbors choosing the price they want. Consumers can benefit of buying clean electricity cheaper than from the retailer. They propose other different application to be integrated in the platform (for electric vehicles, neo-retailers...)	Mix of Public and Permissioned blockchain on top of Ethereum; Proof of Stake	ICO launched, 34 M USD raised,	Charges a fee for each P2P transactions and sells platform licenses to retailers
3	TransActive Grid (Brooklyn Micro-grid); USA [31]	Joint venture between the start-up LO3 and ConsenSys. Pilot project in Brooklyn, where prosumers inside a micro-grid can sell their surplus energy to neighbors. Consumers specify how much they want to buy each day from the clean energy of the microgrid and from the central grid. Based on the use of the platform they are also incentivized	Ethereum. More info not found	In pilot project	Consumers must pay a service charge
4	Conjoule; Germany	They provide a P2P marketplace platform in Blockchain	Ethereum. More info not found	No ICO, raised 4.5 M USD in	No information available

	[32]	where prosumers and consumers from the same location can interact. They also provide forecasts of consumption and production so that users can make their assumptions.		funding round. Pilot projects in Germany	
5	Drift; USA [33]	It provides a platform in which consumers and prosumers choose their energy mix. Drift is building and maintaining an independent network with DERs and power plants from which they buy energy to supply it to their consumers. They use models to forecast user demand and weather. In case demand is higher than supply they resort to the high-frequency trading.	Does not use Blockchain but DLT	Founded in 2014 Raised over 10 M USD in total. Already constituted as a utility and retailer in the USA (May 2017).	They charge 1 USD subscription fee to consumers. They say they offer their users wholesale market price and not the retail one. They also pay their suppliers weekly. They assure they save money in reducing administrative and banks interest fees.
6	Jouliette; Netherlands [34]	Platform in which users in a smart-grid community in Amsterdam can buy clean energy using the token Jouliette. The application incorporates machine learning for forecasts. With Jouliettes, members can pay for car sharing or at local shops and bars.	No information available	In September 2017 the token was created. They are now working on incorporating new features to the platform as the car sharing.	No information available
7	Dajie; Italy [35]	Use an IoT device connected to the smart meter of the consumer and Dajie's blockchain platform. For every kW/h that it is generated by the PV of a user, he receives a token that can be exchanged with other neighbors so that they can use it to buy electricity from their supplier (also connected to Dajie). It is a way to incentivize the production of clean energy	No information available	Building the platform	Charges a subscription fee to consumers of 5 € per month and 20-30 € per year and per customer to energy companies
8	MyBit [36]	They provide a crowdfunding platform for users to invest in shared	Ethereum.	ICO launched. Raised 2.8 M USD. Building	It retains a 1 % of the revenues of the

		solar panels. They obtain return when their energy is sold in P2P markets or to the grid. Payments are done in real time. They want to extend the use of the platform to 3D printers, self-driving cars, drones and more.		the Alpha of the Dapp	energy sold from the PV panels
9	Solar Coin [37]	They give incentives to PV generation. PV owners receive 1 SolarCoin each MWh they produce. These coins can be redeemed by services or goods of participant merchants or exchanged by fiat currencies	Public Blockchain similar to Bitcoin; Proof of Stake and Proof of Solar Energy generation	Fully operational	The validators of transactions receive a 2% interest for their effort. SolarCoin charge a fee in the transaction of SolarCoins for the redemption of services or goods.
10	WePower [38]	It enables green energy project financing. Large scale power plants and smaller in the future, sell in the platform in advance the energy they will generate at a lower price than the future forecasts to be able to finance its construction. They want to be a legal agent being able to buy and sell electricity at the wholesale market, so that their suppliers/consumers can use the tokens for selling/buying energy from them	Ethereum	ICO launched. Raised 40 M USD. They are building and testing the platform and preparing the connection to the network infrastructure.	They charge 0.9 % of the clean energy sold by the suppliers. This energy is tokenized and distributed between all token holders. WePower will keep 30 % of the overall of the tokens, so they will get around 0.3% of the energy generated.

*Table 2. Current Blockchain projects for Energy Communities*

In Table 2, entries 1 to 7 are projects that are developing solutions similar to the one presented in this thesis to allow P2P energy transactions in a local market. The ones that are more developed and that are getting more popularity are Power Ledger in Australia and the Brooklyn Microgrid in the USA. This last one, was the pioneer and first one to use Blockchain for energy transactions. Pylon Network has also been traced closely throughout the development of the thesis because it is deployed in Spain. All these projects were revised and studied to come up with the final solution exposed in this document.

Entries 8 to 10 in Table 2 are not P2P energy trading projects. However, they are about investing in PV panels and incentivizing their installation and generation of clean energy. They were also useful for defining the final idea of the prototype presented in the thesis.

It is also important to mention that due to the early phase of development of this technology and the fame that it is having, over the one year in which the thesis was written new projects aroused and the ones mentioned developed a lot. In the same way, it is important that the reader understands that in one year time from now, the scope of projects might be completely different with some of the ones shown on the table being obsolete and other new ones gaining a lot of popularity.

## Chapter 5 LOCAL ENERGY COMMUNITY

### GOVERNED BY A BLOCKCHAIN PLATFORM

After having explained the main aspects of energy communities and Blockchain technology, it is time to put both concepts together and get deeper into the solution developed in this thesis.

This chapter defines and explains how a local energy community managed by a Blockchain platform looks like and what its main features are. The main advantages and disadvantages of this kind of system will also be described as well as the hypotheses and parameters used for the setup of the prototype built during the thesis and whose main parts will be depicted in the following chapters.

#### 5.1 DESCRIPTION OF THE PROPOSED BLOCKCHAIN GOVERNED ENERGY COMMUNITY

---

##### 5.1.1 COMPONENTS OF THE COMMUNITY

---

The community configuration taken into consideration for the project is formed by a **group of demands**, usually one or several blocks of apartments, single-family houses inside a residential area or an industrial park, where there is only one connection point to the distribution network. The community will be considered as a pseudo-microgrid since its internal connections are isolated from the outside grid except for one connection point. From the electricity system, the community acts as a unique active consumer. This connection to the distribution network will serve as the balancing point for the community since it will proportionate energy when there is lack of generation and it will absorb energy when there is a surplus of generation.

In each community there will be a figure called **Local Community Bank** or **Owner** that is in charge of the maintenance of the inside network, installing the self-generation technologies, emitting tokens, interacting with the pool market and distributing payments between neighbors. This role can be played by a retailer, but ideally by a cooperative or association formed and governed by the community.

Local energy community governed by a Blockchain platform

Inside the community, there will be **some energy resources** that will be owned by the members of the community, or external investors in their absence, by means of the buying of shares. Examples of these sources of energy can be PV panels, heat pumps, CHP micro cogeneration, wind turbines or batteries. There can also be some technologies that are owned individually such as PVs, batteries or electric vehicles.

All components of the community, i.e. demands and generation technologies, are equipped with **smart meters** connected to the blockchain platform. There is one extra smart meter placed at the entrance of the community to measure the flow of energy entering or exiting it. These “blockchain-aware” smart meters act as the points of validation between the electricity system and the blockchain. They will be signing the transactions of electricity measurement registrations that will become tamper proof and stored in the blockchain.

Below, Figure 11 shows a schema of a simple local energy community governed by a Blockchain platform.

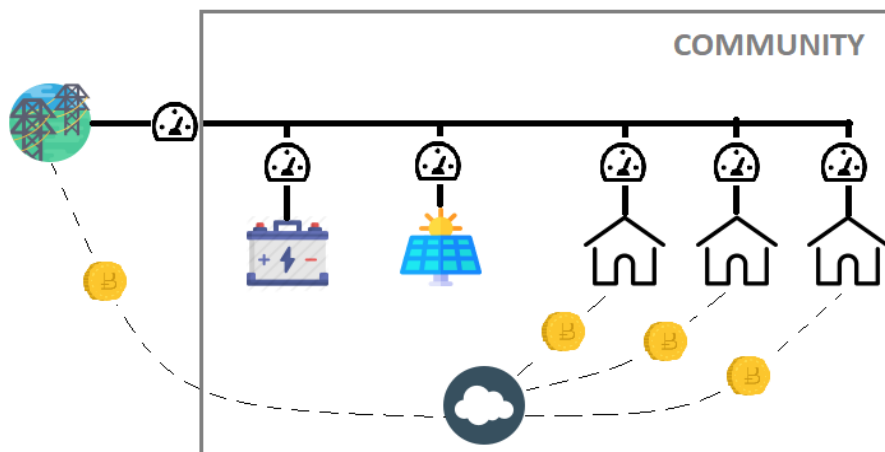


Figure 11. Energy community governed by a Blockchain platform schema

Source: Own elaboration

Looking at Figure 11, the components of the community are summed up bellow:

- **Demands:** Community members’ households. These members have an account in the blockchain platform for the corresponding economic accountability.
- **Shared self-generation technologies:** PV, batteries, wind turbines, heat pumps etc. that are collectively owned by the members of the community.
- **Smart meters:** They measure and register in the blockchain the flows of energy going through them each hour.



- **Community Bank or Owner:** It distributes payments between members, buys and sells energy at the pool market, installs technologies and emits tokens for the correct functioning of the platform and the financing of the technologies.
- **Electricity pool market and DSO:** The community, via the Owner, buys and sells energy directly at the pool market. It will also pay to the DSO or retailer the corresponding network charges for each hour and the contracted power of the community. If these two agents have an account in the blockchain platform, they will be able to receive payments in tokens, if not, the Owner will be the one in charge of exchanging its tokens to fiat currency<sup>3</sup>.

### 5.1.2 FEATURES AND MAIN PARTS OF THE PLATFORM

---

The platform's main objective is to provide its members with the following features:

- Allowance for investments in shared self-generation technologies.
- Optimization of the election of the needed technologies and their operation.
- Forecasts of future demand, generation and market prices for choosing the correct amount of energy to buy or sell at the pool market.
- Automatic registration of energy measurements in a distributed tamper proof data base.
- Conversion of energy measurements into financial values and recording them into the data base
- Settle financial transactions between users and the Owner depending on the recorded values.

To ensure all these features, the platform is formed by two well distinctive tools. One is in charge for the optimization and forecasting simulations and the other one for the measurements registry and financial transactions inside the Blockchain.

#### 5.1.2.1 *Optimization and forecasting tools*

Optimization tools for dimensioning and operation of the energy community are programmed outside of the Blockchain. As today, it would be inefficient to run big optimization problems with many variables in a distributed computing system. This occurs because every transaction in a blockchain has to be mined (verified by the miners)

---

<sup>3</sup> Fiat currency: Any currency declared by a government to be legal tender. For example Euros and Dollars.

and uploaded to every node of the network. Transactions that change a lot of values or data stored in the Blockchain will have to pay a bigger fee so that to be mined.

To keep it efficient, it is preferred to solve this kind of optimization problems by centralized platforms and using languages that are more appropriate and intended to them like Matlab or GAMS.

Optimizations are run in two different phases of the community:

First, for the sizing of **initial configurations** of the community such as the contracted power, contracted tariff and most important the dimensions of the self-generation technologies to be installed. In order to do so, the tool has to take into consideration the number of users inside the community as well as data of their demands and irradiation in the area where the community is located.

If any of these parameters drastically changes, especially if there are new users joining or leaving the Energy Community, new forecasts and optimizations have to be run to see if the initial configurations should be modified. For example, if five new neighbors join the energy community, it may be necessary to install some new solar panel cells to have an optimized generation-demand balance.

Second, forecasts are needed for the **daily operation** of the platform, more specifically, to know what amount of energy should be bought or sold for next day in the energy market. Therefore, every day, the optimization tool has to make predictions on the following 24 hours demand, irradiation and day ahead market prices. In addition, every hour, based on incoming measurements and more precise forecasts of next hours, there should be an adjustment in the operation of the resources to prevent big deviations from the daily prediction and thus from the energy bought for the day.

The tool used for the Energy Community prototype created for the optimization simulations is called OBAMA which is a GUI of Matlab. Chapter 6 fully explains the tool in detail.



### ***5.1.2.2 Blockchain platform for the energy community***

As already mentioned before, smart meters register every hour their energy measurements into the Blockchain. These measurements are transformed into financial figures for their accountability and future payments. In addition, there has to be a record of the shares that each member has in the self-generation technologies.

A new environment is created in which users have to be able to buy and sell shares in technologies between them, pay for their electricity bills and earn money for the energy generated depending on their ownership percentages. Blockchain is the perfect suitable

platform for this new situation since it permits the digital transfer of value without the need of an intermediary solving the double spending problem.

For this environment, a **dual token system**<sup>4</sup> has been adopted. Table 3 shows the characteristics of the two tokens.

<p>EC coin (Energy Community Coin)</p> 	<ul style="list-style-type: none"> <li>• It is the utility token of the platform, i.e. it acts as the gas of the platform. Without it, users cannot operate in the platform. It can also be seen as a limited license for its use, since once you finish your balance of EC coins you need more to be able to keep using the platform.</li> <li>• It is the coin in which payments corresponding to the electricity bill and earnings depending on the contribution to generation are done.</li> <li>• It also serves for buying SCP tokens</li> <li>• It can be exchanged for fiat money or other crypto currencies in exchange houses.</li> <li>• Users will also receive incentives in form of EC coins for the production of green energy and for a specific respond in demand when needed.</li> </ul>
<p>SCP token (Self-Consumption Participation token)</p> 	<ul style="list-style-type: none"> <li>• It is an asset based token. It represents a share in the generation capacity of the self-</li> </ul>

<sup>4</sup> The use of this dual token system was inspired by two other blockchain projects: Power Ledger, already described in Part I.Chapter 4Chapter 4 and Smart Lands which uses Asset Based tokens for representing shares of farms and agricultural lands.

	<p>consumption technologies. Each community has a different one.</p> <ul style="list-style-type: none"> <li>• It is listed in a private market and not in crypto exchange houses. Its value will likely decrease with the depreciation of the assets.</li> <li>• It gives rights to its holders to the energy generated by the delivery of economic benefits in form of EC coins</li> <li>• The Local Community Bank emits it for funding the installation of the technologies. In case of insufficient funds obtained inside the community, it will be allowed for external investors to buy it.</li> <li>• It can be bought or sold from other members giving the user liquidity on their assets.</li> </ul>
--	--

Table 3. Dual token system

Figure 12 shows the flow of value inside the Energy Community for a better understanding of the transactions in which each token can be involved in.

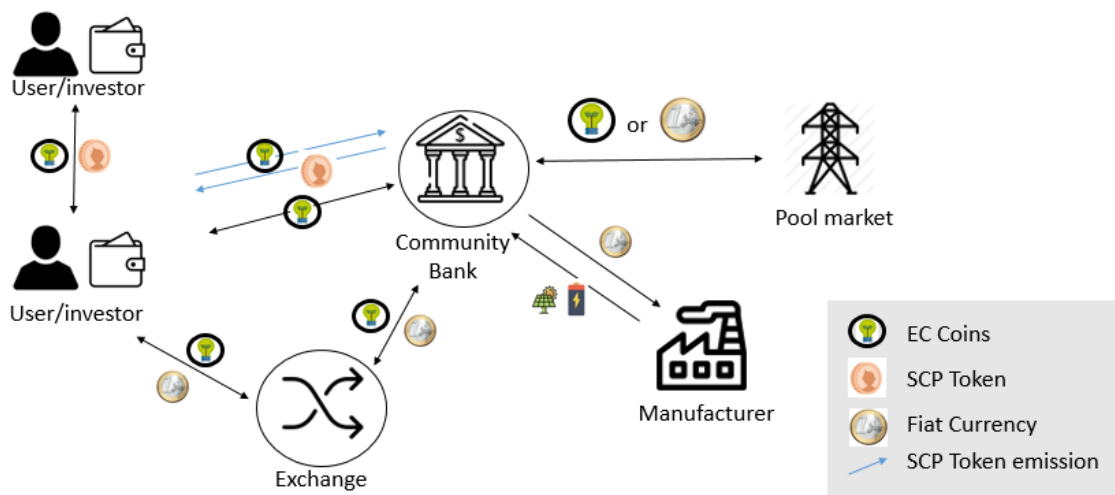


Figure 12. Flow of value inside the community

Source: Own elaboration

Below, the main interactions that can be seen in the value flow map of Figure 12 are described.

- The Community Bank buys some self-consumption technologies to the manufacturer by paying with fiat currency.
- The Community Bank emits some SCP tokens for the funding of the technologies. Investors buy them with EC coins
- Investors, users and the Community Bank can commercialized EC coins for fiat money in crypto currency exchanges
- Investors and user can buy and sell SCP tokens between them by using EC coins
- The Community Bank pays for energy to the pool market with fiat currency or with EC coins if it is integrated in the platform.

### 5.1.3 FUNCTIONING OF THE PLATFORM

---

After having described the main features of the Energy Community platform, this section will go through the steps of its process of operation.

1. **Optimal investments:** First of all, using the optimization tool, it is determined the optimal technologies, and their sizes, to be installed. Based on this, the number of SCP tokens to be emitted, to achieve the required funds, are calculated. These tokens have an associated yield that is notified to the investors. With time, depending on the evolution of the community, their value will deviate from the initial predictions.
2. **Daily pool market:** Every day, following forecasts of demand, solar radiation and market prices, the Community Bank will buy or sell energy for the next 24 hours. The Spanish electricity market system has been chosen as the one in which the platform is operating. The tariff of a Spanish consumer is composed of two main terms: 1) A fixed term or contracted power term that refers to the maximum instantaneous power that can be consumed by the user. The price of the contracted power in €/kW is defined by the DSO. 2) A variable term that refers to the energy that is used by the consumer in an established period of time. The price in €/kWh is composed of different terms. The main term is the market pool price that was defined by OMIE at the specific hour, then it is added the access tariff in which the consumer pays to the DSO the electricity providing service. In addition, there are other lower terms, in which the consumer pay for services such as interruptibility, adjustment of prices and energy, capacity of the network, regulator taxes and others. For simplicity, in this paper, all of these terms are put together into one that is called “network taxes”. Each of the three terms can be seen in Table 4 below.

3. **Energy measurements recording:** Every hour, it is registered all the measurements coming from the smart meters of the Energy Community, as well as the energy market prices. In Table 4, it is depicted the terms recorded in the Blockchain each hour.

Term registered	Abbreviation	Specifications
User demand	$d_i$	<ul style="list-style-type: none"> <li>• Always positive</li> <li>• In kWh</li> </ul>
Generation (sum of PV, wind turbine, CHP etc.)	$Gen$	<ul style="list-style-type: none"> <li>• Always Positive</li> <li>• In kWh</li> </ul>
Battery	$Batt$	<ul style="list-style-type: none"> <li>• Positive when discharging; Negative when charging</li> <li>• In kWh</li> </ul>
Community entrance	$Comm$	<ul style="list-style-type: none"> <li>• Positive when energy entering; Negative when energy exiting</li> <li>• In kWh</li> </ul>
Pool market price	$P_{pool}$	<ul style="list-style-type: none"> <li>• In €/kWh</li> </ul>
Access tariff	$P_{access}$	<ul style="list-style-type: none"> <li>• In €/kWh</li> </ul>
Network taxes	$P_{network}$	<ul style="list-style-type: none"> <li>• In €/kWh</li> </ul>

*Table 4. Terms registered every hour*

4. **Hourly Payments and earnings calculations:** After having registered all terms shown in Table 4, the equations below are used to calculate the final value of the financial transactions each user has to make each hour corresponding to the energy flows.

Equation 1 shows the calculation of the buying price ( $P_{buy}^h$ ) in hour  $h$ . This price refers to the cost that one kWh has in hour  $h$  when receiving it from the external grid.

$$P_{buy}^h = P_{pool}^h + P_{access}^h + P_{network}^h$$

*Equation 1. Buying price calculation*

Equation 2 shows the calculation of the selling price ( $P_{sell}^h$ ) in hour  $h$ . this parameter specifies the money the community gets when selling 1 kWh in hour  $h$ .

$$P_{sell}^h = P_{pool}^h - P_{network}^h$$

*Equation 2. Selling Price calculation*

When selling energy, users will still have to pay for the network taxes. Therefore, they will get the market pool price minus the taxes they have to pay. Logically, the buying price will always be more expensive than the selling one, since the community will have to pay for the access tariff only when buying and the network taxes have to be paid either if they buy or sell.

Equation 3 shows the calculation of the money entering or exiting the community at hour  $h$  ( $M^h$ ). If the energy is entering the community, the buying price will be used and  $M^h$  will be positive. On the other hand, if the energy is exiting the community the price being used will be the selling one and  $M^h$  will be negative due to the sign of  $comm^h$ .

$$\text{if } comm^h > 0; \quad M^h = comm^h * P_{buy}^h$$

$$\text{if } comm^h < 0; \quad M^h = comm^h * P_{sell}^h$$

*Equation 3. Money entering or exiting the community calculation*

Next, in Equation 4, the savings the community gets in hour  $h$  are calculated.

$$\text{if } comm^h > 0; \quad Savings^h = \left( \sum_N d_i^h - comm^h \right) * P_{buy}^h$$

$$\text{if } comm^h < 0; \quad Savings^h = \sum_N d_i^h * P_{buy}^h + comm^h * P_{sell}^h$$

*Equation 4. Community savings calculation in hour  $h$*

Where:

- $N$  is the number of users in the community.
- $(\sum_N d_i^h - comm^h)$  is the difference between total demand and the energy entering the community, i.e. the energy that is being supplied by internal generation coming from the shared self-consumption technologies.

When energy is exiting the community, all demand is being satisfied by the internal generation.

Finally, with Equation 5, the financial balance in hour  $h$  for each user is calculated. The user will get a percentage of the savings of the community depending on the amount of SCP tokens he/she has.

$$Balance_i^h = Savings^h * \%SCP\ tokens - d_i^h * P_{buy}^h \quad \forall i$$

*Equation 5. Internal balance calculation*

5. **Balance registration:** each hour, the balance calculated as in Equation 5 is registered into the blockchain and added to the accumulated total balance of previous hours for every user. The same thing is done for the total balance of the community or  $M^h$  calculated in Equation 3. These balances are recorded in EC, so at the time in which the registration is done, the value should be converted from € to EC using the price at that hour.
6. **Settlement:** A settlement period is defined. For example one month. Every settlement period, the balances of each user as well as the total one of the community are settled. In addition, during the settlement, the fixed term of the contracted power has to be also paid for that period. The following economic transactions between agents are generated:
  - If the user balance is negative, a transaction with that value in EC coins will be executed from the user to the Community Bank. However if it is positive, the transaction will have the opposite direction.
  - One transaction per user to the Community Bank paying for their corresponding percentage in the contracted power. Usually with the installation of self-consumption technologies, the contracted power will diminish. Each user will pay the initial contribution they had in the power contracted before the installation minus a percentage of savings originated for reducing the fixed term, depending on the number of SCP tokens the user has. Equation 6 shows the calculation of the final percentage of the fixed term each user has to pay.



$$\%P_i^{new} = \frac{\%P_i^{old} * P_{cont}^{old} - (P_{cont}^{old} - P_{cont}^{new}) * \%SCP \text{ tokens}}{P_{cont}^{new}} * 100$$

*Equation 6. Calculation of power percentage to pay by each user*

Where  $\%P_i^{new}$  is the new percentage of the fixed term, user  $i$  has to pay.  $\%P_i^{old}$  is the percentage he paid before when there were no self-consumption technologies.  $P_{cont}^{old}$  and  $P_{cont}^{new}$  are the contracted power before and after the installation.

- When all the fixed term payments are gathered by the Community Bank, it will redirect it to the DSO.
- If it applies, the Community Bank delivers incentives for green production and for demand response to the users.
- If it applies, the Community Bank will have to pay for deviations in the market.
- If it applies, the Community Bank will charge a fee in the transactions to users to cover for possible deviations and for the green incentives.

As it could be appreciated when reading through this chapter, internal prices of the community are set taking as reference the pool market prices. More specifically there are two prices:  $P_{buy}^h$  and  $P_{sell}^h$ . The first one is used for all the energy used to cover demand, while the selling price only takes action for the energy sold to the exterior of the community. Therefore, all internal transactions have the price of  $P_{buy}^h$ , which means that generated energy is given the highest value possible in that hour, since if it was set higher, users without generation would be buying the energy from the outside grid at a cheaper price instead of that one. The highest possible price is applied so that to incentivize users to invest in their self-consumption technologies.

In other words, prices are set in the fairest way thinking of all members, and referenced by the pool market. It was preferred to develop a solution in which prices are calculated automatically based on the electricity market rather than creating a private market inside the Energy Community since the pool price is calculated out of complex algorithms based on perfect competition of a lot of agents, i.e. it is the “perfect price”. It is also believed that giving the power to the users to bid on the electricity they are going to buy each hour is a tedious task that takes time and that not everyone would be willing to do.

## 5.2 MAIN ADVANTAGES OF A BLOCKCHAIN ENERGY COMMUNITY

---

This section presents the main advantages that members of an energy community governed by a blockchain platform may have.

- Direct economic benefits for users and investors: Members of the community, for the only reason of being part of it will have two direct economic benefits: 1) they will see a reduction in their electricity bill contracted capacity payment, due to the simultaneity factor. In other words, the sum of each individual contracted power will always be less than the total community power contracted divided by the number of users. 2) They will be able to get incentives for the appropriate demand response when needed.

Investors (members or external people from the community that own SCP tokens) will see a reduction in their electricity bills in both the variable and fixed terms, when their resources produce electricity. There will also be a record of the amount of green energy they are generating for possible future incentives.

- Thanks to Blockchain technology, transparency is given to members. They do not have to put their trust anymore in any centralized institution as distribution or retailing companies for a correct read of measurements, electricity bill deduction, investments in renewable energy and other tasks. With this solution based on Blockchain they just know that the rules followed are the ones written in the protocol.
- Thanks to smart contracts, the logic can be programmed and processes can be automated, eliminating intermediary agents and reducing costs.
- With this solution a new way of investments via virtual tokens is possible. It allows members to invest small quantities of money in self-consumption technologies shares in contrast with having to make a big investment on buying an entire one for their own.
- In addition, with the SCP tokens, investors have a higher liquidity of their assets as they can sell them anytime they want.
- A tamper-proof renewable generation accountability can be achieved for future incentives campaigns run by governmental institutions or others.
- Micropayments become feasible due to the fact that no payments intermediary such as banks are needed anymore. This allows for fast and no fee transactions.

- Having all data stored in a distributed ledger increases security since at least half of the nodes would need to be hacked in order to introduce any malware. This is computationally practically impossible at the moment. Therefore, members' financial accounts will be safe.
- Energy Community users will have access to all the information about their financials and their energy bill statistics through an App for learning about them and changing habits that cost them money and had not noticed before.
- Users may reach the feeling of being part of a community that fights against the climate change and does it in a high new technological way. Being part of an energy community not only brings economic benefits to its members but also social and environmental ones. [18]

### 5.3 MAIN LIMITATIONS AND RISKS OF USING BLOCKCHAIN FOR ENERGY APPLICATIONS

---

---

Although Blockchain has many advantages as described in previous section and it has the potential to disrupt the energy sector, it also faces some limitations or risks that might slow down its widespread use.

First, public and permissionless blockchains that are the most popular, proven type and secure such as Bitcoin or Ethereum have high cost and slow speed transactions. These blockchains use PoW consensus (explained in section 2.2), which requires high computation of miners to create a new block and add it to the blockchain. While payments technologies as Visa can run 4,000 transactions per second in average, Bitcoin can only process a maximum of 7 since the blocks are limited to 1 MB [39]. In addition, to execute a Bitcoin transaction can consume over 5,000 times more than a credit card one, or which is the same, the same electricity needed to power 1.6 households for a whole day [40]. Figure 13 shows a comparison of power consumption per transaction and per year of Bitcoin and Ethereum contrasting it with Visa and residential consumption of certain countries.

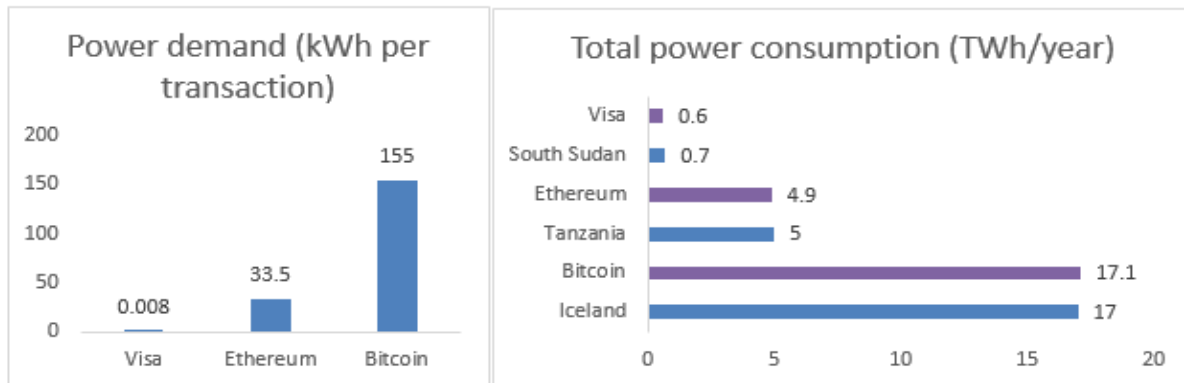


Figure 13. Bitcoin and Ethereum consumption of electricity in contrast with residential consumption in chosen countries

Source: [41]

Second, blockchains lack of flexibility after they are deployed to the network. For making significant upgrades into the blockchain, it is needed a big percentage of stakeholders buying-in. If there are strong popular disagreements inside the community, it can lead to creation of forks. The threat of forking is higher when there exists the possibility to own physical assets via the blockchain since it could lead to a duplication of the same right to a share in a solar panel. [23]

Third, although blockchains look very safe for their intrinsic characteristic of being decentralized and immutable, they are programmed by human beings and they can contain some bugs that might go unnoticed at the beginning because they are not attractive for cyber-attackers. If this bug persists in time, and users start putting money into the blockchain, it could get stolen and difficult to trace. For example, in 2016 an unknown attacker was able to steal 50 million USD from an Ethereum application due to faulty code. [42]

Fourth, users are responsible to keep the private keys that give them access to their virtual accounts. If they somehow lose them, they will not be able by any mean to recover the funds they had.

Fifth, privacy is questionable for public blockchains since everyone can see what are all the transactions and funds that an account has although no one knows to whom that account belongs. There is a bit of skepticism about if this private information sharing that public blockchains bring will comply with the GDPR regulations that come into effect in 2018 in the EU. [40]

Sixth, the operation of the current transmission and distribution networks are natural monopolies. They are in charge of operating, maintaining and supervising the network and its agents. Every new P2P energy trading platform appearing will always have to

reconcile its surplus or lack of energy with the corresponding DSO or TSO. They will not be able to work completely independently as long as they are connected to the central grid. This makes difficult the reduction of intermediaries and therefore decentralization.

Seventh, in a similar way, tokens that represent physical assets as self-consumption technologies need for the approval of the legal and social system to be made effective, and therefore central authorities must be needed regarding this aspect as well.

Eighth and last, an important number of Oracles (defined in section 2.2) must be needed for an application of Blockchain in the Energy sector for accessing outside information such as market prices or sun radiation from public websites or centralized institutions. If this information is wrong or hacked before entering the blockchain, it will cause incorrect process of transactions that will be hard or impossible to recover.

#### **5.4 SUITABLE TYPES OF BLOCKCHAIN AND CONSENSUS**

---

To make effective the maximum amount of advantages discussed in section 5.2 and to prevent as much as possible the risks and limitations described in section 5.3 hereafter it is presented what type of blockchain and consensus are the most suitable for an application such as the one being developed in this project.

First it can be stated that the slow processing latency that characterizes PoW consensus protocol makes it impossible for an application such as this to scale. In contrast with Bitcoin where punctual transactions are made to pass value, in an Energy Community, cyclical transactions have to be processed. For example, every hour there is a need to carry out some transactions for the record of measurements. If there was to be a big number of smart meters connected to the blockchain, the wave of transactions entering at the same time would collapse the network.

Furthermore, it would be counterproductive to be using a system that consumes such a big amount of electricity for an initiative that fights for the implementation of renewables and increase of energy efficiency.

Therefore, other consensus that do not use mining work are more suitable. PoS is the one in which more development has been made. Even Vitalik Buterin, creator of Ethereum is already working on moving its network to a PoS protocol [43]. Other projects working on P2P energy trading as Power Ledger are also using PoS. [30]

It would also be possible to create and implement new protocols that fit better to the specific application to adopt. For instance, Pylon Network project is developing what they

called Proof of Collaboration (PoC). Inside their blockchain they have Certificated Nodes of Validation (CNV). The protocol chooses which the next CNV to form next block is. In order for the chosen CNV to be able to create it, each other CNV in the network has to give its approval. [44]

Regarding the type of blockchain to use, most current projects in development are setting their protocols and Dapps on top of Ethereum Blockchain (see Figure 4) so on top of a public blockchain. It is comfortable for them to do it this way because they do not have to worry about building up a whole infrastructure, and furthermore, they can assure to their users that the system is secure and stable.

However, public blockchains may confront privacy issues as commented in section 5.3. This and other fears such as not having a superior institution to which direct when problems with your account or possible failures in the system occur, opens the opportunity to private or permissioned blockchains.

The main limitation of private and permissioned blockchains is security since they are usually small and do not have a big community under it for correcting code, improving protocols and more.

Nevertheless some initiatives that are getting importance such as the private blockchain **HyperLedger** or the permissioned blockchain **Alastria** in Spain which is backed up by the most important national institutions and that complies with all European legislation. These kind of projects open a new range of possibilities to enabling energy blockchain applications to succeed and become scalable.

## 5.5 PARAMETERS AND HYPOTHESES FOR THE DEVELOPED PROTOTYPE

---

This section presents the Energy Community configuration that has been used, its most important parameters and the main hypotheses taken to keep the prototype simple and feasible according to the resources had.

The **configuration** of the Energy community is shown in Table 5.

Variable	Quantity	Explanation
Number of users	3	Real anonymized data of three single-family house demands located in Valencia (Spain) for one full year, 2017, with a granularity of one hour was used

Number of Self-consumption technologies	1 PV and 1 Battery	PV and batteries are the most frequently used technologies for domestic self-consumption and with only three users it made no sense to take wind turbines into consideration due to their big installation cost.
Number of smart meters	6	One for each user, one for the PV, one for the battery and one for the entrance of the community.
Number of virtual wallets	5	Three corresponding to the users, one to the Owner or Community Bank and one that will represent the pool market and DSO. For simplicity there will be one single account that will take payments related to the buying of energy and contracted power. From now on this agent will be called DSO.

Table 5. Configuration of the Energy Community of the prototype

The main hypotheses taken to develop the prototype are listed below:

*Hypothesis 1:* Due to the lack of the integration of APIs into the platform that connect to external websites, it is not possible to acquire the prices of the electricity market or the cryptocurrencies in real time. This was done this way to keep the prototype simple. The owner will be the one in charge of setting manually the prices of the EC coins and SCP tokens. Regarding the electricity market prices, since simulations are using data from 2017, they will be taken from the ones that took place then, and entered from an Excel sheet every hour as if that was the API. The conversion of prices from € to ETH will be maintained to 1 ETH = 1,000 € during all simulations.

*Hypothesis 2:* For the same reason as before, the Owner will be in charge of refreshing the contracted power and its price manually when required.

*Hypothesis 3:* Due to the lack of devices that emulate the behavior of smart meters and are able to sign transactions in the blockchain, their measurements will also be uploaded to the network via an Excel sheet, in the same way as the electricity prices. Every hour it will be input the corresponding measurement from the Excel sheet to the smart contract. The Owner will be the responsible for signing these transactions.

*Hypothesis 4:* There is not going to be any deviation between forecasts and real measurements. Data used for the optimization simulations that would determine the forecasts of the Energy Community is the data that is actually going to be introduced as real measurements when running the Dapp simulations. This enables the use of the Excel sheet as commented in previous hypotheses to substitute the use of APIs or smart meter devices. Since forecasts are perfect and there are no deviations, there will not be any adjustment costs that the Community has to take care of. Furthermore, for this reason and for the sake of simplicity, financial transactions done to buy energy from the pool market will be executed at the settlement phase as well as the rest of the transactions, instead of each 24 hours.

*Hypothesis 5:* There will not be any extra incentives for the generation of renewable energy, nor for the appropriate demand response of users. This, as well as Hypothesis 4 are left as future work.

*Hypothesis 6:* Due to what it is stated in Hypotheses 4 and 5, and since there is no extra cost for the Community Bank in form of deviations or incentives, it will not charge any fee in the users transactions. No maintenance costs for the self-consumption technologies will be assumed either.

*Hypothesis 7:* No specific regulations regarding shared self-consumption, sun tax or possibility to congregate in a community so that only the external smart meter is measured by the operator, are taken into consideration. This way the best case scenario of the Energy Community can be studied.

*Hypothesis 8:* In the same field as the previous hypothesis, it will not be taken into account any regulation regarding the possibility of using physical assets based tokens.

Next, hypotheses related to the self-consumption technology parameters are stated:

*Hypothesis 9:* Regarding the type of battery, lithium-ion batteries have been chosen due to their level of development and their optimistic projection of costs. It has been considered a useful life of 6,000 cycles with a depth of discharge of 90% and an efficiency of 95%. This will correspond to a useful life of approximately 10 years.

*Hypothesis 10:* Regarding the type of PV, monocrystalline silicon panels have been chosen since they are the most frequently used. It will be installed with an inclination of 35° and oriented to the South. Regarding the efficiency  $\eta_{pv}$ , whose calculation is shown in section 6.2.3, the value chosen for the simulations has been 0.85.

*Hypothesis 11:* Self-consumption technologies are straight line depreciated to 0 to the end of their estimated useful life. It will be considered 20 years for PV and 10 years for batteries [45].



*Hypothesis 12:* the investment costs that have been taken into account can be seen in Table 6 and having chosen guided by reference [45].

Battery	250 €/kWh
Inverter and installation	700 €/kW
PV panels	1000 €/kW <sub>p</sub>

*Table 6. Investment costs for the installation of the technologies*



## Chapter 6      **OPTIMIZATION SIMULATION TOOL:**

# **OBAMA**

In this chapter, it is explained how the simulation tool works and what has been the process for running the simulations to achieve the best sizing of the technologies to be applied into the Energy Community.

### **6.1 OBAMA**

---

OBAMA stands for Optimal Battery Management. It is a GUI developed with Matlab by the IIT (Instituto de Investigación Tecnológica) from Universidad Pontificia Comillas for the Spanish utility company Gas Natural Fenosa.

This tool optimizes the operation of a hybrid system, electric and thermal, with various generation sources and loads connected to the same node. Some of the parameters that can be included are: batteries, PV panels, electric vehicles, conventional generators, wind turbines, electric demand, thermal demand, cogeneration and thermal storage. [46]

In this project the tool will only be used with batteries and PV panels apart from the electric demand.

The introduction of data is done with an Excel template where all the information regarding the technologies is introduced as well as other parameters related to the simulation.

When the Excel is completed, it is imported to the Matlab GUI and the simulations are run. The optimization problem minimizes an objective function containing all costs from the components as well as other penalization costs. The results are output in a CSV file containing every hour optimal energy flows of each component inside the time horizon selected.

Figure 14 shows the input data introduced in OBAMA in case of only having batteries and PV panels and the results it outputs.

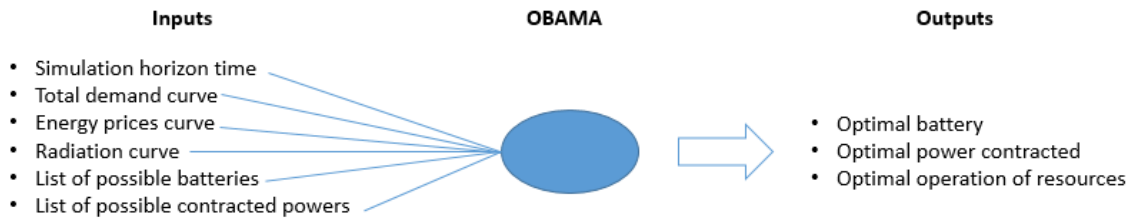


Figure 14. Inputs and outputs of OBAMA

Source: Own elaboration

## 6.2 METHODOLOGY

In order to know which data to introduce to the simulation tool, a previous analysis needs to be done. For the sake of doing this analysis in a structured way and usable in future projects, a methodology was built up.

For the development of the methodology the most important aspect that was considered was the sizing of the battery depending on the three sources of benefit that it can add. First, the reduction of fixed cost term by decreasing the contracted power by shaving the higher peaks. Second, the reduction of the variable cost term by doing price arbitrage if there exists an important variability between different periods in the curve of prices. Third, if self-generation, maximize the value of the energy generated.

Following this criteria the next steps were drawn out:

### 6.2.1 DEMAND CURVE ANALYSIS

It is good to make the analysis of the demand curve for one full year. It is considered that this period of time is enough to provide representative data and it also shows possible different behaviors between seasons.

For the case of a Local Energy Community, since the technologies are to be installed for the benefit of all members, the demand to analyze is the addition of the curves of all members.

When looking at the demand curve, the following variables should be analyzed:

- **Peak-valley distance:** It provides information about the quantity of energy that can be moved from higher to lower demanding periods. In addition, it is important to pay attention to the duration of the peaks and check if there is a common pattern in most of the days, or if in the contrary, every day is different.

- Contracted power: It gives information of what can the peak reduction be when flattening the curve. In order to allocate the total community contracted power into the users in a reasonable way, the highest points of demand can be checked and analyze what the percentages of each user are in each of them. This way, it can be drawn out which users are demanding more power in the highest peaks and therefore should pay a bigger percentage of the fixed term.

### 6.2.2 PRICES CURVE ANALYSIS

The curve of prices is going to be determined by the regulatory scenario in which the demand is located and the tariff contracted by the client. Normally, there is going to be one fixed cost related to the contracted power and a variable one related to the amount of energy consumed.

The objective here is to minimize both costs. Therefore, it is important to identify whether the periods with high demand match the ones with high prices. If this is the case, which is in fact the most frequent scenario, then the strategy will just be to move the energy from the peaks to the valleys.

### 6.2.3 RADIATION CURVE

If there are PV panels included in the analysis, then it is necessary to obtain the radiation curve for the location in which the demand is located. In order to do so, an online tool called PVGIS that can be found in reference [47] was used.

This tool generates average irradiation curves for one day with granularity of 15 minutes for the specified location and month. Other inputs have to be introduced such as the orientation and inclination of the installed PV.

The results of irradiation are given in W/m<sup>2</sup>. In order to obtain the generated power that comes out of the PV, Equation 7 can be applied.

$$E_{pv} = P_{pv,peak} \cdot \frac{Gd}{1000} \cdot \eta_{pv}$$

Equation 7. Calculation of generated power of the PV

Where:

- $E_{pv}$  is the power generation output of the PV panel in kW
- $P_{pv,peak}$  is the peak power of the PV panel installed in kWp.
- $Gd$  is the irradiation value coming from PVGIS tool for each time step in W/m<sup>2</sup>.

- $1000 \text{ W/m}^2$  refers to the maximum irradiation that the mono crystalline silicon PV can accept.
- $\eta_{pv}$  is the efficiency of the PV installation and depends on the losses due to changes in temperature, dirt that can be accumulated on the surface of the panel, shades, electric losses and in the inverter. Normal values to use for this term vary between 0.75 and 0.85.

## 6.2.4 PV AND BATTERIES PRELIMINARY SIZING

When there is only batteries installed, if the peaks of energy match the peaks of prices as said before, the strategy for sizing them will be to find the amount of energy that can be moved from spike to valley to flat the curve as much as possible. When doing this, it will always be important to not use a very big size that would suppose an inconsistent case both economically and physically.

Figure 15 shows an example of how the use of a battery can flats the demand curve transforming it from the red one to the blue one.

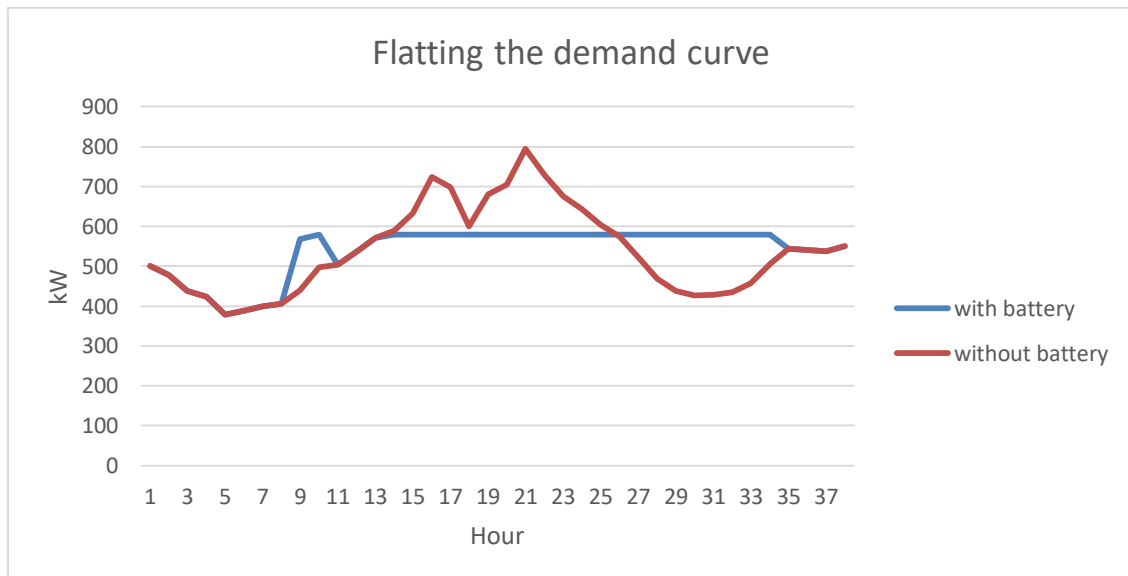


Figure 15. The use of a battery flats the demand curve

Source: Own elaboration

If in the analysis there are also PV panels, the strategy for sizing the battery changes since a new variable is introduced, the surplus of energy generated. To increase the value of the generated energy as maximum as possible, the battery has to be able to absorb as much surplus of energy as possible, so that the least is wasted. Then, it will be used in the periods where prices are higher and demand is not covered by the PV.

The PV panel will be sized so that there is not a big amount of energy surplus, since usually the price for selling it to the grid, if it is even possible, is usually less.

### **6.2.5 SIMULATIONS WITH OBAMA**

---

After a preliminary sizing of the technologies has been done using previous strategies, data is introduced into OBAMA for simulation. This way, it is possible to make adjustments and try similar sizes until an inflection point with less cost solution is reached.

OBAMA's functioning is based on minimizing an objective costs function. As inputs, it is introduced generation curves, demand, prices, possible sizes of batteries, possible contracted powers and more. The output of the simulations are the optimal power term to contract, size of the battery, and its optimal operation curve

### **6.2.6 COSTS AND BENEFITS ANALYSIS**

---

Using the investment costs of the considered technologies and the annual savings obtained from the simulations, a costs and benefits analysis (CBA) is done to determine if it is worthy to invest in these technologies.

To make simple, and since there are several hypotheses already done through the analysis prior to reaching this point, the CBA is kept simple. A simplified payback period is calculated together with the NPV of the project. If the resulting payback is lower than the useful life of the technologies and the NPV is positive, then the Energy Community should invest on it.

This methodology will be used in section 8.1 of results for the case study presented in the thesis.





## Chapter 7      **DESCRIPTION OF THE DAPP CREATED**

In this chapter it will be explained how the Distributed Application (Dapp) was programmed. There will be some aspects covered such as what tools were used for its development, how the architecture is built, what are the main files and their content, what is the logic behind the code and what were the difficulties encountered during the development.

### **7.1 PROGRAMMING TOOLS USED FOR THE DEVELOPMENT**

---

In this chapter, the programming tools required to build the different parts of the Dapp are briefly explained.

#### **7.1.1 NPM (NODE PACKAGE MANAGER)**

---

NPM is the package manager for JavaScript, it is the tool used to install needed modules inside the project. It counts with an enormous depository of modules and it also allows the programmer to manage dependencies inside their projects.

#### **7.1.2 SOLIDITY**

---

The Ethereum language code, Solidity, was used for the development of the smart contract of the Dapp. This language was chosen because it is one of the most popular and widespread, which in addition makes it easy to find tutorials and documentation about it.

As it is defined in the Solidity Documentation [48]: “Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM)<sup>5</sup>”. Solidity is a modular language composed of different functions with particular purposes. This is due to how Smart Contracts are structured.

A Smart Contract is formed by different functions (the code) and by the data it stores (the state of its variables, addresses etc.). All this resides at a specific public address in the

---

<sup>5</sup> The EVM or Ethereum Virtual Machine is the runtime environment for smart contracts in Ethereum

Ethereum Blockchain. These functions can be called or invoked from other addresses in the blockchain such as users or APIs and from other functions inside the smart contract.

A Smart Contract function can be called with two main purposes:

1. To change some data inside the Blockchain. Every time a function is called with this purpose, a new transaction is generated and therefore some gas is required to be paid by the address that invoked it.
2. To read an entry inside the Blockchain. No transaction is generated and thus, no fee has to be paid.

Let's take a look to a very simple example to understand better these two features

```
1  pragma solidity ^0.4.0;
2
3  contract SimpleContract {
4      uint myValue;
5
6      function set(uint x) public {
7          myValue = x;
8      }
9
10     function get() public constant returns (uint) {
11         return myValue;
12     }
13 }
```

Figure 16. Simple smart contract

Source: Own elaboration

The first line just indicates the Solidity version in which the code is written. Then the contract begins. It can be seen that it contains two functions. The function 'set' changes the value of the variable 'myValue'. Therefore, when calling this function a specific amount of gas will be spent by the account that makes the call.

On the other hand, the second function, 'get', just returns the value of the variable 'myValue' without making any change inside the Blockchain. That is why calling this function will not have any associated fee.

The keyword 'public', which is written in the declaration of the functions, means that it can be called by any account.

In order to learn more about smart contracts in Solidity and their programming, reference [48] can be revised. It contains the whole documentation of Solidity from the basics to complex examples.

In addition, in order to learn Solidity and be able to build the smart contract of the Dapp presented in this thesis, a very easy follow through tutorial was pursued. It can be found in reference [49].

### 7.1.3 HTML, JAVASCRIPT AND CSS

---

In order to be able to build a web page, the main programming languages needed are HTML, JavaScript and CSS.

- HTML or Hypertext Markup Language is used to define the content of the web page.
- JavaScript is used to define the behavior of the web page. It contains the programmed logic taking place under the interactions with the web page.
- CSS is used to specify the layout of the web page. In the .css file it has to be defined the distribution, color, formatting etc. of the content stated in the .html file.

It is not necessary to have a very deep knowledge of these languages to be able to build an easy working web page.

The HTML, CSS and JavaScript files form what is called the **frontend application**.

In this project, a frontend application was created so that to be able to interact with the smart contract as a frontend user and for demonstration purposes.

As a guide, for any troubles, questions or doubts that came upon, reference [50] was mainly used. Furthermore, in some cases, discussion forums as “stackoverflow.com” were also consulted.

### 7.1.4 GANACHE

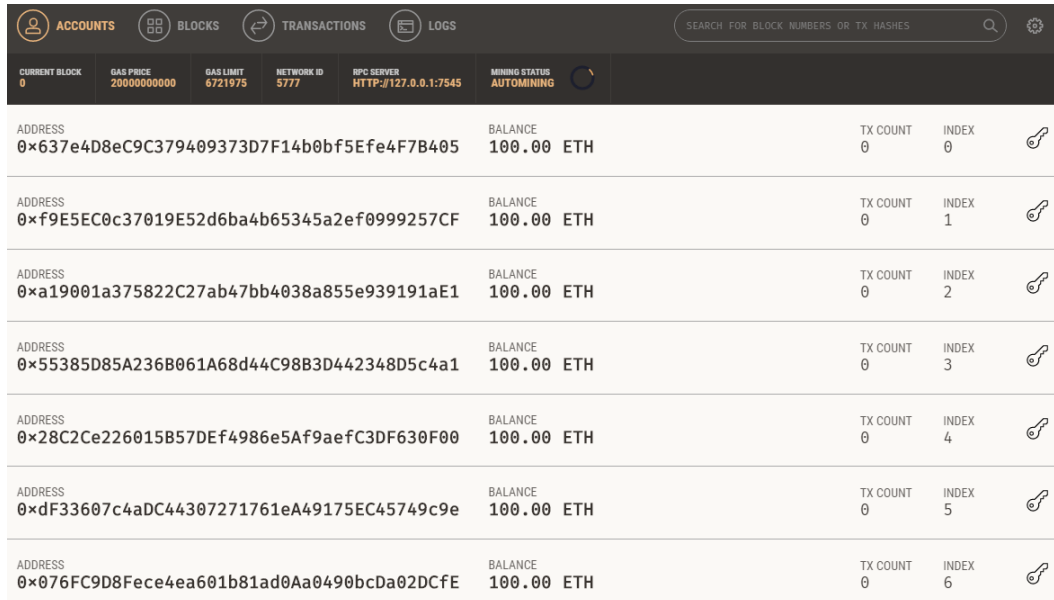
---

A local blockchain built up for testing called Ganache was used. This blockchain has been created by Truffle and can be downloaded from their website [51]. It is easy to use and very intuitive.

Ganache ships with an internal JavaScript implementation of the Ethereum Blockchain where no clients need to be installed. There is only one node that mines all transactions and operates all accounts. Therefore it does not need high power of computation and transactions are uploaded rapidly. This is why it is perfectly suitable for testing Dapps.

When opening Ganache, it will automatically create ten accounts with 100 fake ethers to be used for testing the Dapp.

In Figure 17, it is shown the Ganache layout upon creation of the blockchain in which it can be observed some of the account created with their initial 100 ethers.



ADDRESS	BALANCE	TX COUNT	INDEX
0x637e4D8eC9C379409373D7F14b0bf5Efe4F7B405	100.00 ETH	0	0
0xf9E5EC0c37019E52d6ba4b65345a2ef0999257CF	100.00 ETH	0	1
0xa19001a375822C27ab47bb4038a855e939191aE1	100.00 ETH	0	2
0x55385D85A236B061A68d44C98B3D442348D5c4a1	100.00 ETH	0	3
0x28C2Ce226015B57DEf4986e5Af9aefC3DF630F00	100.00 ETH	0	4
0xdf33607c4aDC44307271761eA49175EC45749c9e	100.00 ETH	0	5
0x076FC9D8Fece4ea601b81ad0Aa0490bcDa02DCfE	100.00 ETH	0	6

Figure 17. Ganache layout

Evidently, testing the Dapp in the main network of Ethereum makes no sense since it would end up in a big economic cost resulting from the numerous transaction fees paid every time the smart contract is deployed or its functions tested.

For the sake of testing, apart from local software as Ganache, there exist some online test Ethereum networks such as ‘Ropsten’, ‘Koban’ or ‘Rinkeby’.

### 7.1.5 WEB3 AND METAMASK

In order to have a web app that is able to interact with the Ethereum blockchain, i.e. having a JavaScript file interfacing with the smart contracts of the Blockchain when a user interacts with the front end application, it is needed to have installed what is called Web 3.

The concept of Web 3 comes with the evolution of the Internet and it can be defined as a new version or extension in which stored data in this kind of web can be understood and processed by machines. So it is related with artificial intelligence and the appropriate communication of webs and systems regarding the interests of users.

For our interests and to be able to make the Dapp work it is needed to provide Web3 somehow. While Web3 is provided, the code written in the JavaScript file will be able to

interact with the smart contract inside the blockchain and work properly. There are three different ways or methods to provide web3 to the Dapp.

1. Main network provider: Web3 is provided by the main network of Ethereum. This means that the execution environment is connected to the node in the real network. This will be used when the developer wants to make use of smart contracts inside the main network. Since the Dapp here presented is run in a local environment, this provider cannot be used. If the Dapp was to be audited and deployed in the Main Network, then it would make sense to use this kind of provider.
2. Inject Web3: Web3 is injected by an outside application provider such as 'Mist' or 'MetaMask'. These applications can be used for running executions in the Main Network but also in Test Networks and Local Networks such as 'Ganache'.
3. JavaScript VM: The execution environment does not connect to any node. Everything is done locally and inside the memory of the PC

In this project, for testing and running the Dapp, 'MetaMask' was used. It is a Google Chrome extension that injects web3 to the application. Every time a transaction has to be submitted by one of the accounts that form the blockchain (the accounts defined by Ganache), the MetaMask window pops up in order to allow it by the signing of the transaction.

MetaMask was used because it is one of the most frequent applications used for signing transactions online and it was perfectly suitable for the Dapp built in this project using Truffle.

More about MetaMask can be found in reference [52].

In Figure 18 it is shown an example of the MetaMask pop up window that appears when a transaction needs to be signed by the account that is creating it. In this case, Account 1 is submitting a transaction in which he is issuing some SCP tokens. He is transferring some data of 68 bytes to the address of the smart contract and a value of 0 ETH. The gas that has to be paid by Account 1 for this action is \$0.11.

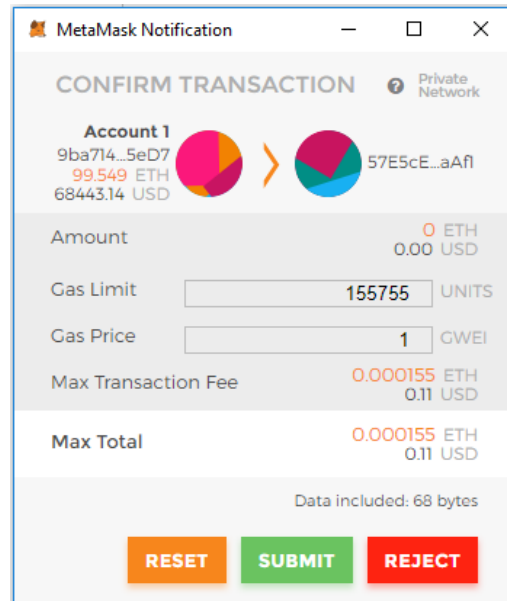


Figure 18. Metamask pop up window

### 7.1.6 TRUFFLE

As defined in their website [51], “*Truffle is a development environment, testing framework and asset pipeline for Ethereum, aiming to make life as an Ethereum developer easier*”.

Truffle provides the developer with an initial library with the needed npm modules and directories needed for writing a Dapp. It gives the coder a starting point so that he can focus more in the content of the important files and less in the architecture behind that connects them.

Furthermore, in their website, they have what they called ‘*boxes*’, which are more advanced initial architectures that combine other modules so that the user can choose what is more appropriate for the kind of Dapp he is going to build. Some of them even provide an example of a Dapp so that it is more intuitive for the programmer and he can understand better the main features of the box.

For the development of the Dapp for the Energy Community, the ‘*webpack box*’ was chosen, which includes contracts, migrations, tests, user interface and webpack build pipeline. In the following section, 7.2, the different parts of the library and their functions are described with more detailed.

This box was selected because it offered an initial library with all suitable parts to build up a simple Dapp. There are other libraries with more advanced tools and modules that offer a better design of the app and more shortcuts or facilities in the programming. However, due to the lack of knowledge of these programming languages and tools, it was preferred to stick to basic JavaScript and HTML.

When creating the project with the mentioned Truffle Box, these are the commands available for the developer:

- Truffle compile: This command will compile the smart contracts, i.e. the files with extension .sol, and will indicate any error encountered.
- Truffle migrate: This command will run the migration files that will deploy the smart contracts to the blockchain. In order to be able to do this, evidently, the .sol must be first compiled and show no errors.
- Truffle test: Used for running tests that try out the different functions of the solidity file to check if they are working as the developer thought so.
- Npm run dev: This command will run the web app, and a new window in the browser will open as localhost showing the front end of the app.

It is also worthy to mention that in Truffle website [51], they have a tutorial called '*pet-shop*' that was very helpful for understanding the functioning and the main content of each of the files of a Dapp based on a Truffle library as well as learn how to use web3 features to interact with the smart contract from the JavaScript.

## 7.2 ARCHITECTURE OF THE DAPP

---

The final architecture of the Dapp is shown in Figure 19. Some files were originated from the Truffle initialization and were not touched in any moment. Others were fully developed by the author. In the next sections, the main folders and files are explained. The more relevant will be described with more depth but about others for the sake of brevity, it will just be mentioned what their function or mission is.

In the The electricity sector is undergoing big changes in the last decade. Global warming is fostering the introduction of renewable energy, there is being an increase of the overall demand, digitalization is increasing, consumers are becoming more active and electricity services are being introduced at a distributed level. In addition, consumers are shaping to be more interested in services than in products and the sharing economy is at its greatest

splendor. Parallel to this, Blockchain technology appears. Blockchain is able to provide transparency, security, efficiency and cut costs of digital transactions based solutions and it is called to be a technology that disrupt many different businesses.

This thesis presents the main advances that have been done in the area of research and the main projects that are establishing Blockchain for P2P energy trading. It discusses the main advantages and limitations that the technology has to succeed and become scalable and finally it explains the delivered prototype consisting of a Blockchain platform that manages the internal transactions of an energy community. This platform is able to analyze the forecasts of demand curves of participants, prices curves and weather conditions and optimize the dimensions of the generation technologies to be used as well as their operation. In addition, it allows participants to invest in those technologies by buying digital tokens that can be exchanged or sold at any time. It keeps an immutable record of the measurements coming from smart meters and it couples the energy spending and generating to economic transactions that are as well automatically executed between members. Ultimately, with this platform, users can become more aware and participative in electricity services inside the community in a transparent and trustworthy way, they can invest in renewable energy in a more liquid and simple manner and they can socially and economically contribute to the energy transition towards a cleaner planet. Other applications of Blockchain to the electricity sector seem also promising to reduce transaction costs and integrate environmental-friendly solutions.

Furthermore, a necessary step to achieve an energy transition towards a decarbonized economy is to revise market designs, regulations and ultimately economic signals provided to the different stakeholders of the electricity sector.



Annexes, the full code for the more important files is shown.

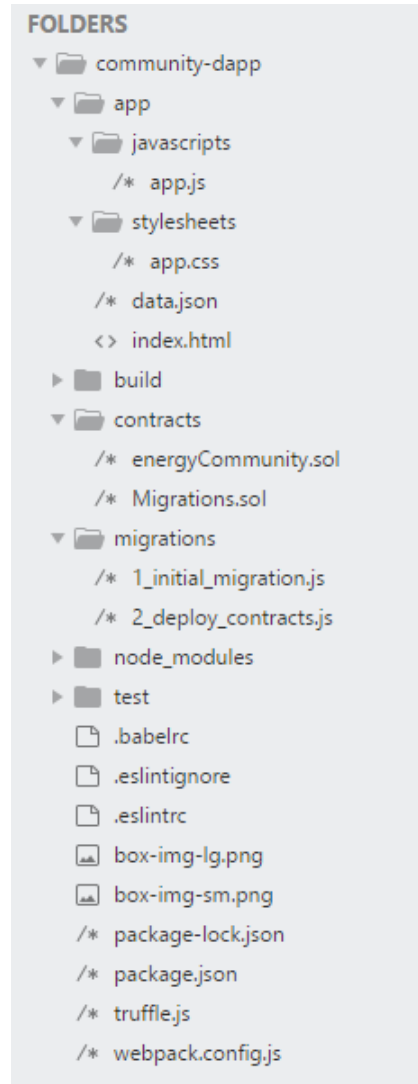


Figure 19. Architecture of the Dapp

Source: Own elaboration

### 7.2.1 UPPER LEVEL FOLDER

In the upper level folder, i.e. community-dapp folder, it can be found several files that are not contained in any subfolders. `.babelrc`, `.eslintignore`, `.eslintrc` and `package-lock.json` are predetermined files that were created directly with the initialization of the project with Truffle Webpack box and that were not touched in any moment throughout

the development of the Dapp. For the sake of simplicity and briefness, they will not be explained here.

On the other hand it is worth to mention the meaning that have the last three files that appear in Figure 19:

### **7.2.1.1 *Package.json***

This file serves as the index of the project as it states information about the libraries used. Every time the project is initialized, the file refreshes the information with the new dependencies installed.

It shows the creator of the initial Truffle library, its version, the license, the main scripts and all the development dependencies installed in the project.

The '*package.json*' file serves as a guide, so that when the project is seen by other developers, they know what libraries and languages are being used and how the architecture makes sense.

The code for this file is shown in Annex A. Code in section Part I.10.1.110.1.1

### **7.2.1.2 *Truffle.js***

Truffle.js is the configuration file of the Truffle project. Its content was created with the initialization of the box and it was neither touched throughout the Dapp development.

In the file it is specified the network the Dapp will connect to, its port and its host. In the Energy Community Dapp, these values correspond to the ones of the Ganache GUI that is the private local network in which the smart contract is deployed.

The code in '*truffle.js*' is shown in Annex A. Code in section 10.1.2.

### **7.2.1.3 *Webpack.Config.js***

This file is characteristic of Webpack module. Every time the programmer executes the command '*npm run dev*' (already mentioned in section 7.1.6) this file is run. Its mission is to combine all the Front-End application files, so all the JavaScripts, the HTML and the CSS, into one single JavaScript with all the code converted to be compatible with old and new browsers. This is very convenient for developers because they can use new features of programming languages without having to check if the browser supports them.

In the case of this project, this file, which code can be checked in section 10.1.3, combines the three files that form the Front-End application: '*app.js*', '*index.html*' and '*app.css*' constituting the resulting one called '*app.js*' that will reside in folder 'build'.

The code inside this file was already given when creating the project with the Truffle Box as well. However, some adjustments were made so that it was compatible with the files existing in this project.

## **7.2.2 CONTRACTS FOLDER**

---

This folder contains the Solidity files of the project.

### ***7.2.2.1 Migrations.sol***

This smart contract also comes predetermined with the installation of the Truffle Box and it contains some initial settings required for the correct functioning of the deployment of the subsequent written smart contracts to the network. It is needed so that the Migrations (see section 7.2.4) can be used. This contract will be deployed once as the first migration and will not be updated again. The code of this file is presented in Annex A, section 10.1.4.

### ***7.2.2.2 energyCommunity.sol***

This is the main file of the project. It contains the smart contract created for the automatic registry of measurements of the energy community. In addition, it carries the accountability of the financials of the users inside the community and it runs the corresponding settlements when appropriate. Even further, it allows the addition and removal of users and smart meters, the exchange of tokens between users and everything else needed for the Energy Community to work properly.

The content of this file was entirely developed by the author. The full code can be seen in the annexes in section 10.1.5. It is complemented with comments so that it is easy to follow.

Hereafter, the most important parts and functions of the smart contract are briefly described:

- In the first part of the contract, events and variables are declared and initialized.
- Afterwards, the constructor can be observed. This function is run at the same time as when the contract is deployed and cannot be called again. Inside of it, it is defined the number of EC coins that are emitted and their price. This function is shown in Figure 20. All EC coins created are added to the owner's balance account.

```
//CONSTRUCTOR
function EnergyCommunity (uint _initialPrice, uint _initialSupply) public{
    owner = msg.sender;
    balanceEC[msg.sender] = _initialSupply * decimals; // EC*10e10
    priceOfECCoins = _initialPrice; //in wei
}
}
```

Figure 20. Constructor code

Source: Own elaboration

- The modifier ‘*onlyOwner*’ is declared right after the constructor and before all the rest of the functions. It is used in the functions that can only be invoked by the Owner of the contract, i.e. the creator of the community or also called before in the document the Community Bank
- Then, functions corresponding to the setup of the community are stated. They serve for the following:
  - Add smart meters
  - Remove smart meters
  - Add users
  - Remove users
  - Set the contracted power of the local community and its price

All these functions are only available for the Owner of the community. As an example, in Figure 21, the function used to add users can be seen. It has two inputs that the owner will have to insert when calling the function: 1. the public account address of the user in the blockchain and 2. The ID that will identify the new user in the community. It can also be seen that the modifier ‘*onlyOwner*’ is used. Inside the function, the new user created is added to the array ‘*users*’ that keeps the registry of all the users in the community. An event of new user created is deployed to the network and the variable that keeps track of the number of users is increased by one.

```
function addNewUser(address _addr, uint _id) public onlyOwner{
    users.push(User(_addr, _id, 0, 0));
    NewUserCreated(_addr, _id);

    numberOfUsers ++;
}
```

Figure 21. Function to add users

Source: Own elaboration

For brevity, the rest of the functions mentioned are not explained here. All the code with comments is shown in the Annexes, in section 10.1.5.

- The next part of the smart contract clusters the functions that are related to the operations that can be done with the tokens of the platform. (EC coins and SCP tokens). These functions are briefly explained below:
  - Set price of EC coins. Only the owner can call this function and will change the current price, in Ether, of the EC coins.
  - Buy EC coins. Any user can call this function to buy EC coins from the Community Bank. They will pay Ethers in exchange.
  - Mint SCP tokens. Only declarable by the Owner and is used to emit SCP tokens for the investment in new self-consumption technologies.
  - Buy SCP tokens: Users can buy SCP tokens from the Community Bank in exchange of EC coins.
  - Burn EC coins and Burn SCP coins: Only the Owner can call them. They are used if it exists the need to eliminate tokens.
  - Give permission and Buy SCP tokens from other users: These two functions are used by the users to be able to exchange SCP tokens between them. With the ‘Give Permission’ function, the user liberates some of his SCP tokens so that another user can buy them from his account.
  - Withdraw: This function can only be called by the Owner and it serves to collect all the Ethers that have been paid from the users to the smart contract when buying EC coins.

As an example, in Figure 22, function ‘Buy EC Coins’ is shown. It can be seen that in its declaration there are two key words: 1. ‘public’ that means that anyone can call the function and 2. ‘Payable’ which means that when calling it, some Ethers can be sent to the smart contract account. Afterwards there are two

‘require’ statements to make sure that 1. The number of Ethers sent (*msg.value*) corresponds to the correct price that has to be paid to buy the amount of EC coins requested and 2. The Community Bank still has enough EC coins to provide the quantity requested. Finally, the balances of the user that calls the function (*msg.sender*) and of the Community Bank are adjusted.

```
function buyECCoins (uint _numberOfCoinsToBuy) public payable{
    require(_numberOfCoinsToBuy * priceOfECCoins == msg.value);
    require((_numberOfCoinsToBuy*decimals) < balanceEC[owner]);

    balanceEC[owner] -= _numberOfCoinsToBuy*decimals;
    balanceEC[msg.sender] += _numberOfCoinsToBuy*decimals;
}
```

Figure 22. Buy EC Coins function

Source: Own elaboration

- Following into the smart contract, the next group corresponds to getter functions. These functions are called to look at information residing in the blockchain. They are declared with the keyword ‘view’ so that it is made sure that the state of the blockchain is not modified. As explained before, in chapter 7.1.2, there is not any associated fee when the state remains the same. The getter functions of this smart contract are used by the users to get information about:
  - My EC coins balance
  - My SCP tokens balance
  - My current instantaneous internal balance (balance of EC that has not yet been liquidated) as well as the array containing all values from the past.
  - Instantaneous balance array of the DSO, which is the same as the money entering or exiting the community every hour.
  - History of measurements of my smart meter
  - History of measurements coming from generation, battery or the community smart meters.
- Next, one of the most important functions come. It is named ‘newMeasurement’ and it is called every hour to register new measurements and market prices. Inside this function, the conversion from energy measurement to its financial value takes places as well as the calculation of the community savings and the instantaneous internal balance of each user depending on their demand and their self-consumption participation (SCP tokens). Additionally, this function contains the

code needed to keep track of the maximum power and allocate corresponding percentages of power use to each member for the subsequent settlement.

For simplicity and brevity, the code of this function is not shown here. It can be checked in the Annex, section 10.1.5. Some comments accompany the code for a better understanding.

- Last, the function for settlements of payments is displayed. It should be called once every settlement period (every month for example). For the means of the prototype for the sake of testing, it is triggered by a button. This function generates transactions in EC coins between the Community Bank of the community and the DSO and between the users and the Community Bank, depending on the instantaneous internal balances accumulated. Then these balances are reinitialized to zero. Payments corresponding to contracted power are also settle here.

In the same way as before, the code for this part is displayed in the Annex.

### 7.2.3 BUILD FOLDER

---

As explained in section 7.2.1.3, when the command `'npm run dev'` is executed, the file `'Webpack.config.js'` is run creating a new file that combines all JavaScripts, HTML and CSS of the front-end application. This file will be allocated in the Build folder.

Similarly, when the command `'truffle compile'` is executed, after compiling the Solidity files of the project, a new JSON file is created per each Solidity one containing their ABI (Application Binary Interface). It is indispensable to have the ABI of the Solidity files because it makes it possible for the JavaScripts of the front-end application to interact with the Blockchain. In other words, the JavaScripts interact with the EVM (Ethereum Virtual Machine) or Blockchain via the ABI inside the JSON files and never via the Solidity ones.

These new JSON files are allocated in the Build folder. An example of the execution of the command `'truffle compile'` is shown in Figure 23.

```
PS C:\Users\E\Documents\TFM\solidity smart contracts\community-dapp> truffle compile
Compiling .\contracts\Migrations.sol...
Writing artifacts to .\build\contracts
```

Figure 23. Truffle compile command run on the community Dapp

Source: Own elaboration

## 7.2.4 MIGRATIONS FOLDER

Migrations are JavaScript files that assist the developer to deploy smart contracts into the Ethereum network. They are responsible for staging deployment tasks and they are most useful when only some parts of the code or only some smart contracts want to be tested.

Migrations are a Truffle feature. When having a project with Truffle, the developer deploys its Solidity smart contracts into the network by displaying the command ‘*truffle migrate*’. This will run all the migrations that have not been executed yet.

In this project, there are two migrations files:

### 7.2.4.1 Initial migration

It is needed to deploy the ‘Migrations.sol’ file so that future smart contracts can be run correctly. It will only be run once and will not be updated again

### 7.2.4.2 Deploy contracts migration

This migration is used to deploy the smart contracts of the project. Since in the project in question there is only one, it will be the only execution to make. Figure 24 shows the code for this migration.

```
1 var EnergyCommunity = artifacts.require("./energyCommunity.sol")
2
3 module.exports = function(deployer) {
4   deployer.deploy(EnergyCommunity, web3.toWei(0.0001, 'ether'), 250000);
5 };
```

Figure 24. Deploy contract migration code

Source: Own elaboration

As can be seen in line 4, it is used a command that deploys the smart contract named ‘EnergyCommunity’ and that specifies the inputs for its constructor. So the initial price of EC coins will be 0.0001 Ethers and 25,000 EC coins will be emitted for this community.



In Figure 25, an execution of the command ‘*truffle migrate*’ is shown. It can be seen how the two migrations are deployed and saved into the network.

```
PS C:\Users\E\Documents\TFM\solidity smart contracts\community-dapp> truffle migrate
Using network 'development'.

Running migration: 1_initial_migration.js
  Replacing Migrations...
  ... 0x7abbd22aea638b1165c70a7133f5adce81f21ad93872d9249f3252ceccb886f7
  Migrations: 0x161876bd4e5edbdfafd1b32b6147fae3c0672e6b
  Saving successful migration to network...
  ... 0x5724363d35407e21799e4b49029c72f5f323492aa8191e188d59dbb155d27c23
  Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing EnergyCommunity...
  ... 0x7d6aeb05e30867285f93f74d92fe2c755145d0d9c0892cf74ceb3c1b5a16bf4b
  EnergyCommunity: 0xb1baac9ca9a46bcd6edcc13a1b0613442f6de993
  Saving successful migration to network...
  ... 0x860fd3f048ce398b57519044ec761ddf2334f7daeb56effa39d6c1443b36b09a
  Saving artifacts...
```

Figure 25. Truffle migrate command for the community Dapp

Source: Own elaboration

## 7.2.5 APP FOLDER

This folder contains the files that define the front-end of the application. Additionally, inside it, it can be found the file ‘*data.json*’ described hereafter.

### 7.2.5.1 Data.json

This file contains all the needed data that comes from the Excel originated by the optimization tool (OBAMA) of the community. For the case of the prototype it will then contain the data corresponding to one year of the optimization of the resources (PV and battery) inside the community, as well as the measurements coming from the smart meters of the entrance of the community and the demands of each user.

Data.json will also provide the prices of each hour.

Figure 26, as an example, shows a piece of code of this file with the information of hour 1. Data in kW for the six smart meters of the community are given as well as the prices of that hour.

```
{  
  "hour": "1",  
  "comm": "0",  
  "d1": "0.068",  
  "d2": "0.182",  
  "d3": "0.313",  
  "pv": "0",  
  "bat": "0.563",  
  "omiePrice": "0.05882",  
  "accessTariff": "0.002879",  
  "networkFee": "0.00674"  
},
```

Figure 26. Information for hour 1 in data.json

Source: Own elaboration

### 7.2.5.2 *Index.html and app.CSS*

As already explained before, these two files are in charge of the content of the web page and its style.

On one hand, *Index.html* defines all the content that has to be shown in the application so that the front-end user can interact with it. There are four main different features that can be observed when looking at the web app:

- Headers and fixed labels: Their content is specified in the HTML file and it never changes, it is fixed. They are used to give information to the user.
- Changing labels: Their content is referred to the value of a variable in the JavaScript file. Therefore, it can change over time or after some interaction with the Dapp.
- Inputs: They are empty boxes where the user can write some data to interact with the application
- Buttons: They are another way for the user to interact with the application. By clicking on buttons, some actions are executed inside the app.

To sum up, headers and fixed labels show information to the user of how they can interact with the program. The user can do so by using inputs and buttons and in the end, the state of the app modifications are shown via changing labels.

On the other hand, the *app.css* file defines the style of all the content displayed by the HTML file. By the use of '*div*' the layout of the web page is defined. Each piece of content will be distributed in the different parts of the layout so that the webpage is correctly structured and easy to follow for the user.

Furthermore, each piece of content will have its format defined in the CSS file, including features such as width, height, label, color, transparency and many more.

Full code for index.html and app.css are shown in annexes 10.1.9 and 10.1.10 respectively. These two files were fully developed by the author.

Figure 27 shows the finished web app initial screen.

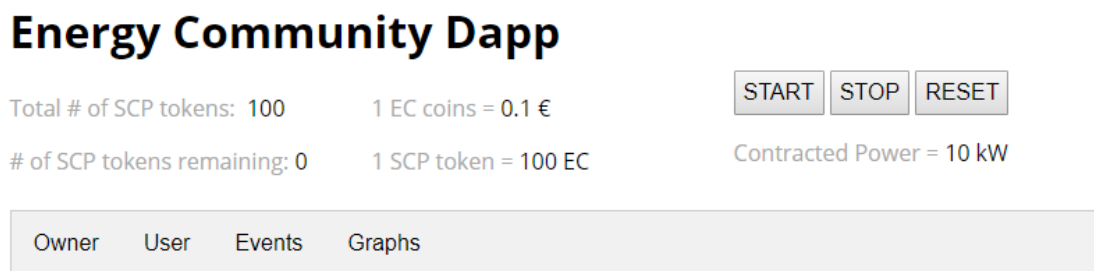


Figure 27. Web App initial screen

Source: Own elaboration

First, a header with the title of the web app is displayed. Below it, it can be seen some important information for all the participants of the community such as number of tokens available, prices of the tokens and the community contracted power. These labels are formed by a fixed part (in grey) and a changing part (in black) that is refreshed every time the value is modified. On the right of the screen, there are three buttons that are used for running, stopping or resetting the simulation. They are used for testing purposes since in reality, the application would be programmed to take inputs every hour nonstop.

In the bottom part of Figure 27, four tabs are displayed that open new windows with more information and features:

1. **Owner:** it shows the main functionalities for the owner of the energy community and it displays information relevant for him. Table 7 shows the different functions that the owner can run with this tab and the main information that is shown to him.

Owner tab	
Functionalities	Information displayed
<ul style="list-style-type: none"> <li>• Emit new SCP Tokens</li> <li>• Set a new price for EC coins</li> <li>• Add a new user</li> <li>• Add a new Smart Meter</li> <li>• Define the address of the DSO</li> <li>• Withdraw money from the smart contract</li> <li>• Burn EC coins<sup>6</sup></li> <li>• Burn SCP tokens</li> <li>• Change the contracted power and its price</li> </ul>	<ul style="list-style-type: none"> <li>• EC coins the owner has</li> <li>• SCP tokens the owner has</li> <li>• What is the current price for contracted power</li> <li>• What is the maximum registration of power since last settlement</li> </ul>

Table 7. Owner's tab functionalities and information

2. **User:** it shows the functionalities available for the users and the important information for them. Table 8 presents what these functions are and what the information displayed in this tab for the user is

User tab	
Functionalities	Information displayed
<ul style="list-style-type: none"> <li>• Buy EC coins from the Owner</li> <li>• Buy SCP tokens from the Owner</li> <li>• Give permission to other user to buy SCP tokens from his account</li> </ul>	<ul style="list-style-type: none"> <li>• EC coins the user has</li> <li>• SCP tokens the user has</li> <li>• Current internal balance of EC the user has since last settlement</li> </ul>

<sup>6</sup> The Owner can burn (destroy) tokens. This can be done for the rising the value of the tokens that are owned by the users since there will be a lower total amount of them.

<ul style="list-style-type: none"> <li>• Buy SCP tokens from other user</li> <li>• Get a table with all the measurements registered by a specific smart meter the user has access to (its demand, or common smart meters)</li> </ul>	
--	--

Table 8. Functionalities and information shown in the different tabs

3. **Events:** it shows all events taking place in the Blockchain. They notify users for important alerts happening inside the energy community such as new introduction of users or smart meters, new incoming measurements or new settlements done. All events are displayed since the origin of the community ordered from newest to oldest. As example, in Figure 28 can be shown the last events at one point of the simulation.

### Events

- NewMarketPrice: PoolPrice= 0.04499€/kWh , AccessTariff= 0.000886€/kWh , NetworkFee= 0.00692€/kWh ; block number: 51
- NewMeasurement: community= 3.872kW , d1= 0.257kW , d2= 0.496kW , d3= 0.119kW , PV= 0kW , Batt= -3kW ; block number: 51
- NewMarketPrice: PoolPrice= 0.04578€/kWh , AccessTariff= 0.000886€/kWh , NetworkFee= 0.00692€/kWh ; block number: 50
- NewMeasurement: community= 1.5989kW , d1= 0.115kW , d2= 0.315kW , d3= 0.1kW , PV= 0kW , Batt= -1.0689kW ; block number: 50
- NewMarketPrice: PoolPrice= 0.0471€/kWh , AccessTariff= 0.000886€/kWh , NetworkFee= 0.00692€/kWh ; block number: 49

Figure 28. Events tab

Source: Own elaboration

4. **Graphs:** This tab basically shows in a visible and understandable way what is really happening each hour inside the community. It displays the measurements obtained in each smart meter (User demand, PV, battery, entry of the community) and the corresponding economic transactions in each hour.

Figure 29 and Figure 30 show an example of the content that this tab displays for User 1 from hours 1 to 24. In Figure 29, it can be observed user 1 demand and the common community measurements corresponding to the self-consumption technologies and the energy community smart meter. Figure 30, shows the economic transactions occurring each hour for the user 1 and the DSO as well as what is the accumulated balance for each of them. All in EC coins.

Plots

1 1 24 plot

Current hour: 48

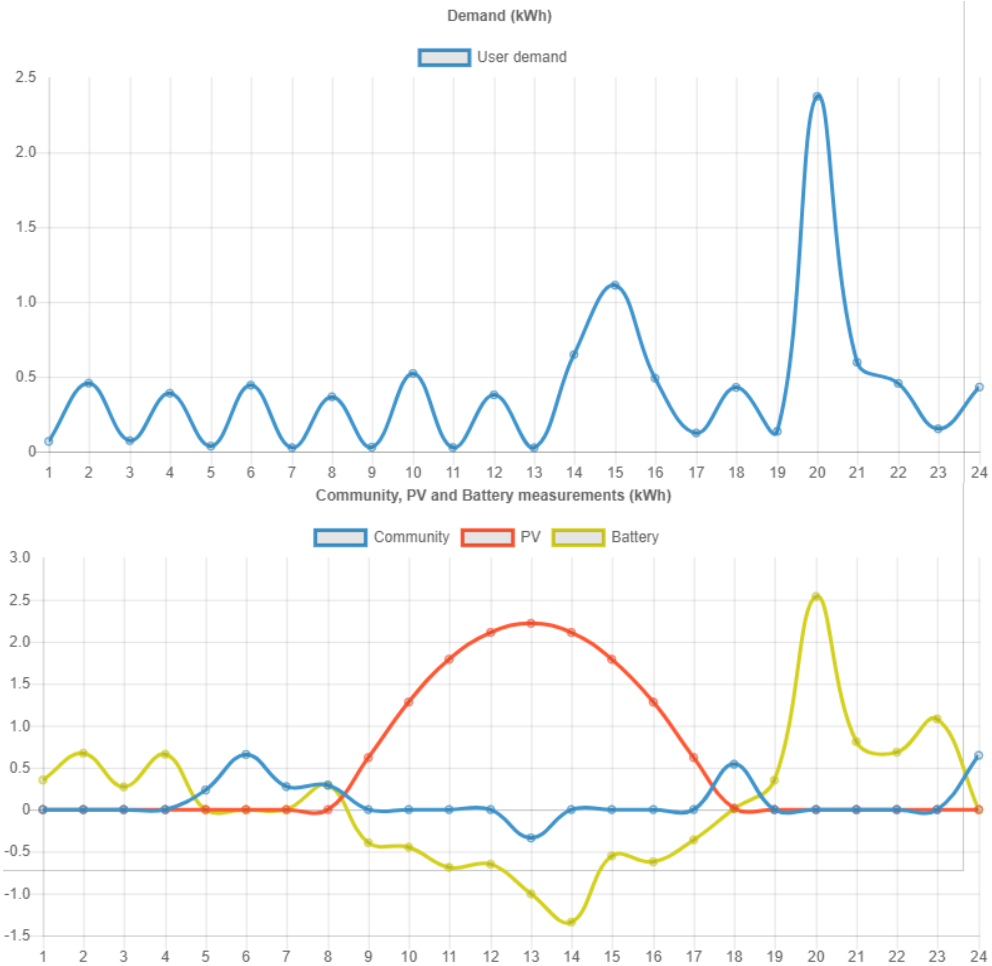


Figure 29. Graphs for measurements of smart meters inside the energy community for user 1 from hours 1 to 24

Source: Own elaboration

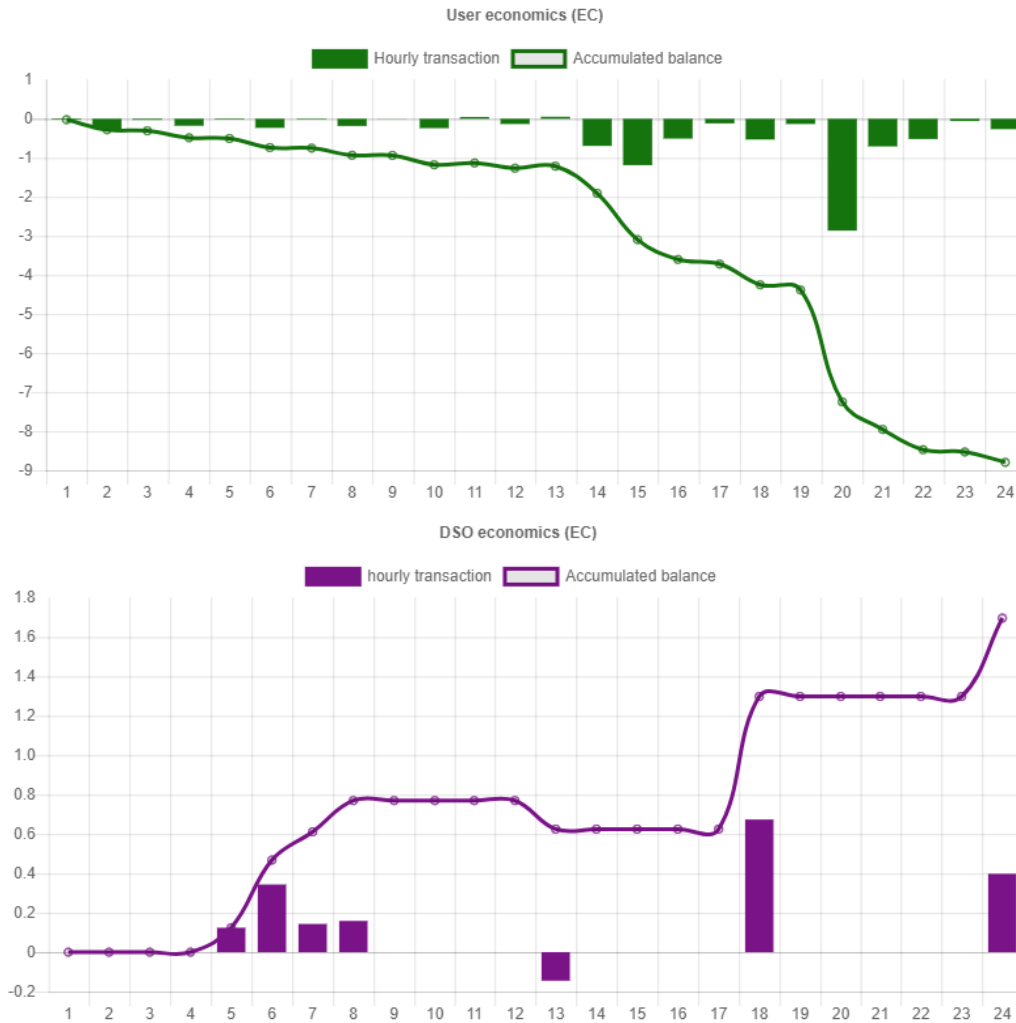


Figure 30. Economic transactions and accumulated balance for the user and DSO

Source: Own elaboration

These statistics may be useful for the users for analysis and improvement of their electricity bill.

More screenshots of the layout of the webpage with examples of possible interactions that the owner or users can do are shown in annex 10.2

### 7.2.5.3 *app.js*

This file is one of the most important of the Dapp. It serves as the nexus of the user's interface and the smart contract residing in the Blockchain.

Additionally, it presents all the logic behind each interaction with the web page. The JavaScript file is listening to events that occur in the HTML such as the click of a button.

When an event takes places, a piece of code with some logic is activated in the JavaScript, sometimes accessing or changing information of the smart contract, and then it sends back the new values to the changing labels in the HTML for their display.

Full code for this file was developed by the author and it is shown in annex 10.1.8. Consecutively, the structure of the document is briefly described from top to bottom for a better comprehension.

- First, libraries that are going to be used throughout the file as web3 and Truffle Contract, as well as the .json file of the smart contract, are imported.
- Global variables are declared and initialized.
- The object ‘App’ is declared and all the methods that have inside are programmed. This part occupies the majority of the document since there is a big amount of methods. Some of them are used for refreshing values in the html and others to make changes in the smart contract following orders coming from the inputs and buttons actioned by the users in the web page. All possible interactions with the smart contract have to be coded and this is why the extension of the file gets so large. Nevertheless, since JavaScript is a modular language (each method, one module), the code is easy to follow and to understand.
- Outside of the object ‘App’, at the end of the file, there is an Event Listener for the reload of the web page that triggers the method ‘start’ of the App and that checks if web3 is already being injected. If there is not, it injects it from an active web3 provider like MetaMask or Mist. If there is none active, an error message is displayed.

Inside the multiple methods that can be found inside the object ‘App’, the features that are most frequently used and that are repeated in most of them are:

- Read values that are either displayed or entered by the user in the HTML interface. This is usually done by the instruction: “*document.getElementById()*”
- Insert values into labels of the HTML. This is usually done by the instruction “*.innerHTML*”
- Insert values into functions inside the Smart Contract. The function is called though a web3 functionality. The calling account, the gas to pay and the inputs to the function can be entered here.
- Read values from public variables or that are returned from functions inside the Smart Contract. Same web3 functionality as before is used to access the smart



contract. Figure 31 shows an example in which the variable ‘*numberOfSmartMeters*’ from the Smart Contract is being accessed.

```
var metal;  
EnergyCommunity.deployed().then(function(instance){  
  metal=instance;  
  return metal.numberOfSmartMeters.call();  
});
```

Figure 31. Accessing the smart contract from the JavaScript file

Source: Own elaboration

## 7.2.6 NODE\_MODULES FOLDER

---

This folder, as indicates its name, contains all the modules installed with NPM. Due to the fact that Truffle was used for the creation of the project, there is a very long list of dependencies and modules existing in this folder so that everything works as a whole. Nevertheless, the programmer has no real need of knowing them and therefore, they will not be explained here since it is out of the scope of the thesis.

## 7.3 LIMITATIONS OF THE DAPP

---

In this section it is presented the limitations that the resulting Dapp has and the aspects in which it should be improved to arrive to a scalable and commercially application. These limitations appear mainly due to some roadblocks that were encountered during the programming and that were not able to be solved. Some of them are originated by the limitations in Solidity or the programming language and others by the lack of needed skills of the programmer.

First, and most important, the Dapp only works for the Energy Community presented in this document with 3 users. If more users are introduced, the Dapp would not work properly. One of the main reasons for this is that due to the lack of devices that emulate blockchain-connected smart meters signing transactions, the measurements were input from an Excel Sheet as previously stated. The problem with this, is that there is a limit of values that can be stored into the stack of the smart contract in Solidity [48]. This means that a limited number of local variables can be declared and/or input into a function. It can vary from 12 to 16 depending on the computation work done inside the function. In the smart contract of the Energy Community created, in the function ‘*newMeasurement*’ eleven values are stored into the stack: nine inputs corresponding to the six smart meters measurements and the three market prices and two more needed local variable declarations. A solution to this would be to split the inputs into more functions, but then

the Owner would have to be signing more than one transaction every time new measurements are uploaded. This would be solved if the smart meters could signed their measurements themselves.

Second, related to the previous one, since every transaction has to be signed, every hour when new measurements arrive, the Owner has to click on the MetaMask pop-up window for verifying the transactions which would be a tedious task in a real life application. Again this would be solved if smart meters could signed these transactions.

Third, MetaMask can only provide the instance of one account at a time. This makes sense in real applications since each user would enter the application from a different computer or smart phone, but for the sake of testing it, it makes it tiresome since every time it is wanted to change from one user's account to another one, the webpage has to be refreshed.

Fourth, the latency for transactions being mined in the blockchain is larger than the web page refreshment one. This means that if changes in the HTML are referenced to interactions done by the webpage user that change the state of the smart contract, the interface will refresh before the variable has updated its value and therefore it will not show. To solve this, either a waiting time can be added or the HTML change can be referenced to the real update of the variable in Solidity. This second solution is difficult to apply with the features available with web3.

Fifth, testing with smart contracts can get tedious since every time it is needed to make a change in the code, the smart contract has to be deployed again and make all the needed initial adjustments to it.

Sixth, during the development of the Dapp it was not possible to achieve a working connection from Excel to JavaScripts so that changes inserted in the Excel can automatically be used by the Dapp. Alternatively, what it was done was to convert the Excel into JSON file and introduce this one into the Dapp.

Seventh, the smart contract code is surely as much depurated and abbreviated as it could get. This is an important feature in Ethereum smart contracts, since each transaction has associated an amount of gas. The more depurated the code is, the less fees the users will have to pay. Nevertheless, as said before in section 5.4, this type of application would be more suitable for a non-PoW consensus in which fees for transaction do not exist.

Eighth, as already commented, Webpack was used in the Dapp to condense all the Front-End files into one. Due to the lack of experience with this feature, it was not possible to work with more than one JavaScript file. The script for it got very large without the possibility of splitting it into different files. This made it difficult to navigate through it due to its big dimensions.

## Chapter 8 CASE STUDY, VALIDATION OF THE PROTOTYPE AND RESULTS

After the Blockchain Energy Community has been ideated together with its configuration, system of tokens, principles of operations and hypotheses, as explained in Chapter 5, the optimization tool has been prepared and its methodology of use defined, Chapter 6, and the Dapp programmed, Chapter 7, it is time to test it and use it. This chapter presents the simulations done with the prototype created together with their results.

First, simulations done with OBAMA for the calculation of the optimized size of the PV panels and batteries are done. These simulations also output annual forecasts of energy flows and prices. Then it is explained how to initialize the Dapp to the Base Case and after that, the results obtained from the simulations are presented.

Figure 32 shows a time line where it is presented the most important events that take place in the operation of the platform.

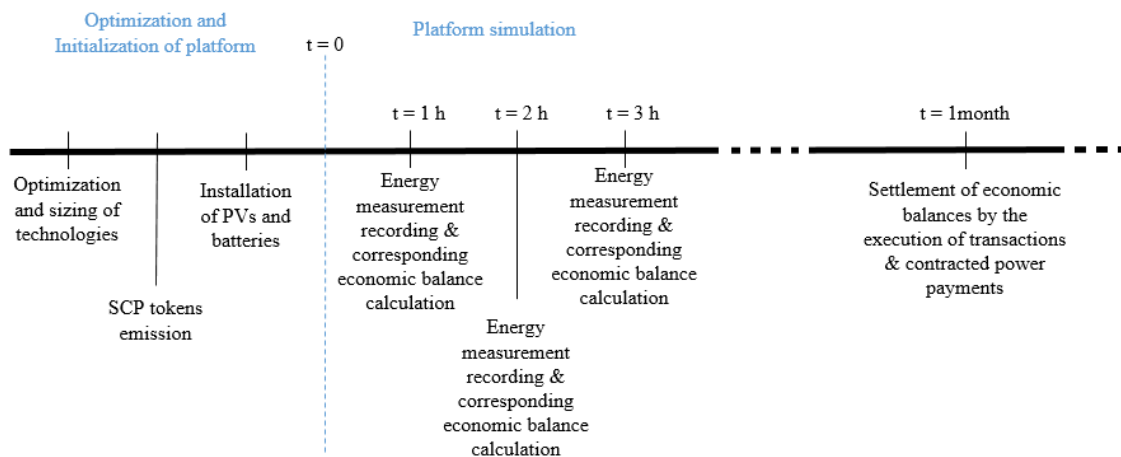


Figure 32. Time line of most important events in the platform

### 8.1 OBAMA SIMULATIONS

For the sizing of the PV panel and the battery, the methodology explained in section 6.2 will be followed.

**Demand curve analysis:**

In Figure 33 it can be seen the demand for the three users of the community for the whole year 2017 starting on the 1<sup>st</sup> of January. User 1 has very constant power peaks throughout the year, while user 2 and user 3 have much higher peaks of demand in winter and or in summer.

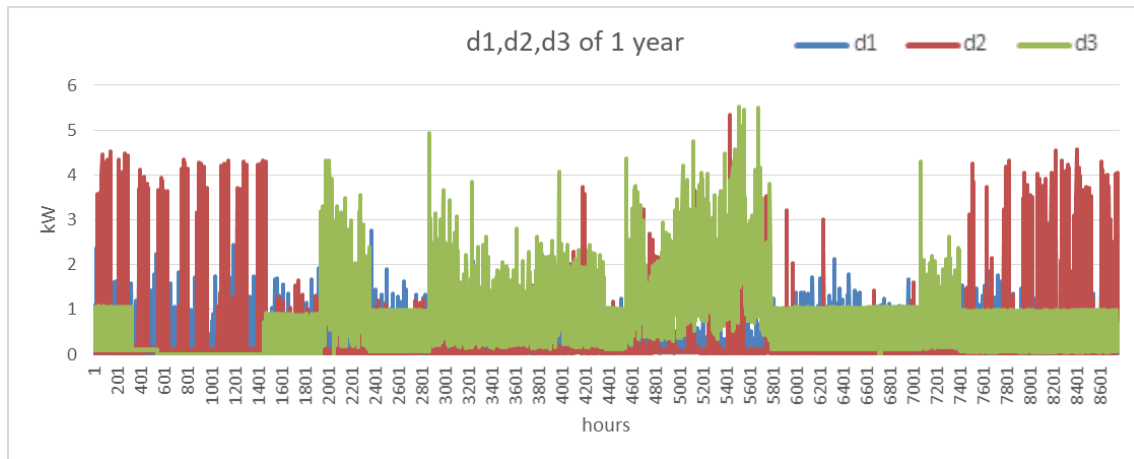


Figure 33. Annual demand of the 3 members of the community

Source: Own elaboration

In Figure 34 it is shown the aggregated demand of the three users. It can be seen that the highest peak of demand is around 8.2 kW which is much less than the addition of the highest demand of each user individually (13.6 kW).

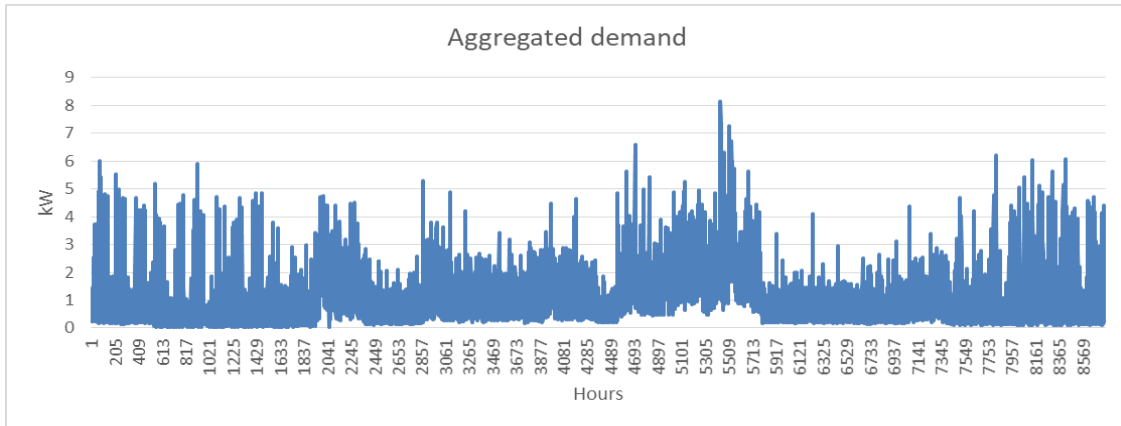


Figure 34. Aggregated demand of the 3 members

Source: Own elaboration

The contracted power of the community is going to be assumed to be 10 kW, leaving some margin from the maximum of the year. In order to allocate the right proportion of the total community contracted power into the users, it is going to be checked the ten points with highest demand and make an average of percentages between them. The final average percentages calculated can be observed in Table 9.

	Power peak average contribution	Percentage
User 1	0.69 kW	6.9 %
User 2	5.137 kW	51.37 %
User 3	4.173 kW	41.73%
Total	10 kW	100%

Table 9. Allocation of power contracted among users

It may look surprising the fact that user 1 has very small contribution into the contracted power payment. This occurs because most of the highest peaks studied, 8 out of 10, took place in one same week in August when user 1 had very few consumption, probably because he was out on vacation.

#### Prices curve analysis:

As already commented in section 5.1.3, the energy community buying price is going to be the curve formed by the addition of the OMIE pool prices of year 2017, the network taxes and the access tariff contracted.

The access tariff that is going to be taken into account for the community is a three period's hourly discrimination tariff in order to be able to take advantage of the PV and batteries as much as possible.

In Figure 35 it can be observed the curve of buying prices for the community during a week in August alongside with the corresponding aggregated demand. It can be appreciated that the majority of the peaks in demand coincide with the periods of higher prices. Therefore, the strategy of moving energy from expensive periods to cheap ones will converge with the one of reducing the contracted power.

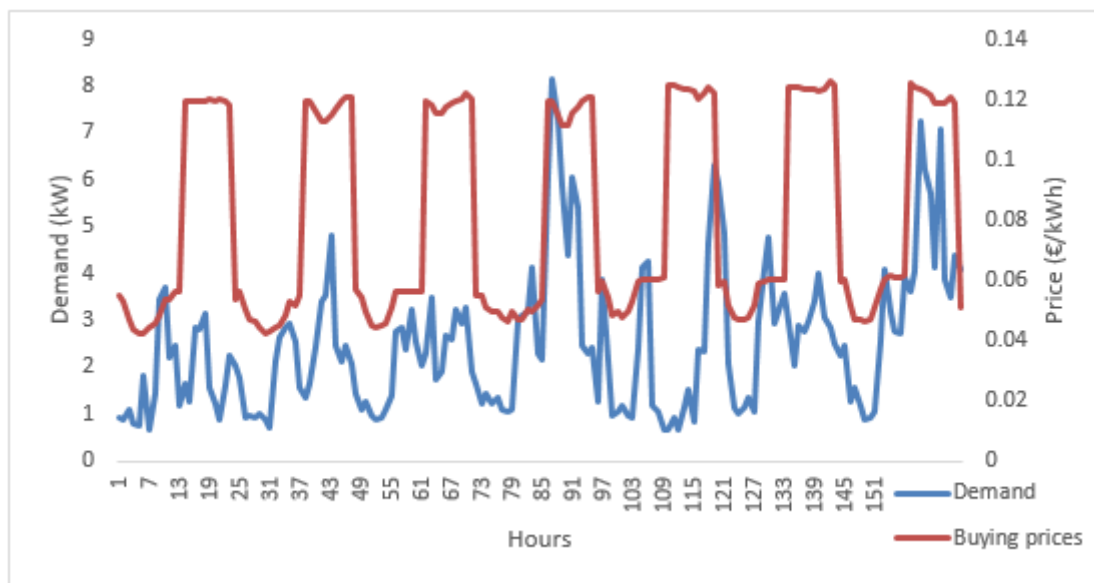


Figure 35. Demand and prices for a week in August

Source: Own elaboration

### Radiation Curve:

The radiation curve is calculated from PVGIS software by introducing the characteristics of the PV panel included in hypothesis 10 and the location of Valencia.

After implementing Equation 7, the resulting curve for the production of energy is presented in Figure 36. The value chosen for the  $P_{pv,peak}$  of the panel installed was 4 kWp. It resulted a reasonable number since it can cover a big portion of demand of higher peaks, usually in summer and winter, and it does not generate such big surpluses in days of low peaks. As an example, Figure 36 shows the curve of generation with the one of demand in a week of August where peaks of demand are higher and in a week of April where there are some surpluses.

Other values for the peak power of the panel were tested such as 2, 6 or 10 kWp. Finally it was decided to use 4 kWp since it produced the best cost-benefits results.

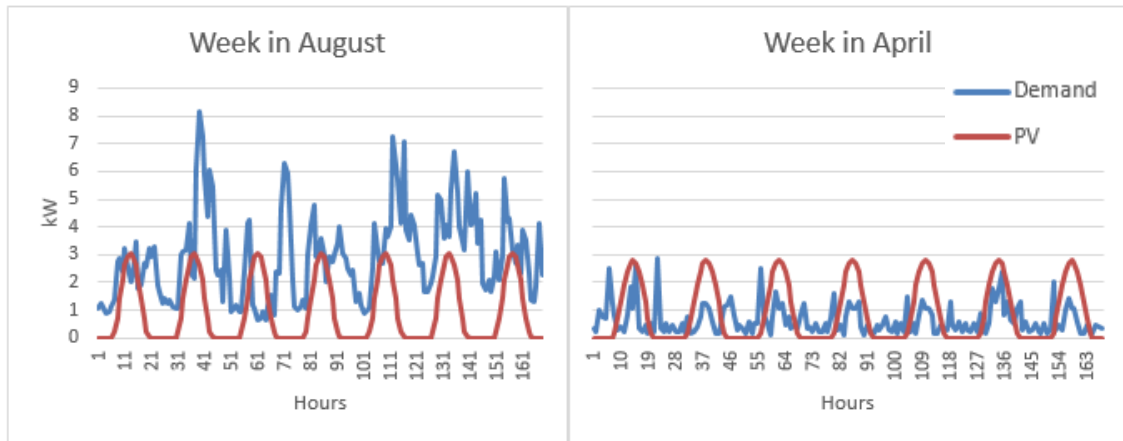


Figure 36. PV and demand curve in August and in April

#### Battery sizing:

For sizing the battery, it is necessary to find a battery that is not too big so that to prevent very high investment costs but that is able to absorb as much surplus energy as possible coming from the PV panel and to reduce the highest peaks for the minimization of the contracted power.

Figure 36 can be again used to see what amounts of energy the battery has to be able to absorb in April and what kind of peaks it has to reduce in August.

#### OBAMA simulations:

After all the previous analysis is done, the curves of demand, prices and PV generation are introduced into the Excel template of OBAMA. In Table 10 the battery sizes chosen for the simulations, with their inverter peak power as well as the possible contracted powers chosen are shown.

Battery 1:	Capacity: 3 kWh Inverter: 1.5 kW	Contracted power 1	3.2 kW
Battery 2:	Capacity: 6.4 kWh Inverter: 3 kW	Contracted power 2	4.6 kW
Battery 3:	Capacity: 10 kWh	Contracted power 3	6.3 kW

	Inverter: 4 kW		
--	----------------	--	--

Table 10. Battery and contracted power inputs of OBAMA

OBAMA, when simulating, crosses all these scenarios and point out which combination of battery and contracted power is the best for the simulating period introduced, i.e. one year.

In this case, the solution elected by OBAMA was the **battery of 6.4 kWh** and the **contracted power of 4.6 kW**.

OBAMA will also output the optimal operation of the battery and the flows of the community with the grid. Figure 37 shows a sample of data of 3 days of April in which it can be seen the final energy flows the community has. The green line shows the energy that is entering in the community. As can be seen, it has been reduced substantially from the original amount that it would be demanding if there were no technologies installed (in blue). It can also be observed that in periods where there is no sun, demand is also covered by the battery since the entering energy is at the 0 line. OBAMA also indicates in which periods the community has to sell energy and how much. Figure 38 shows the resulting state of charge of the battery output from OBAMA for the same time period shown in Figure 37.



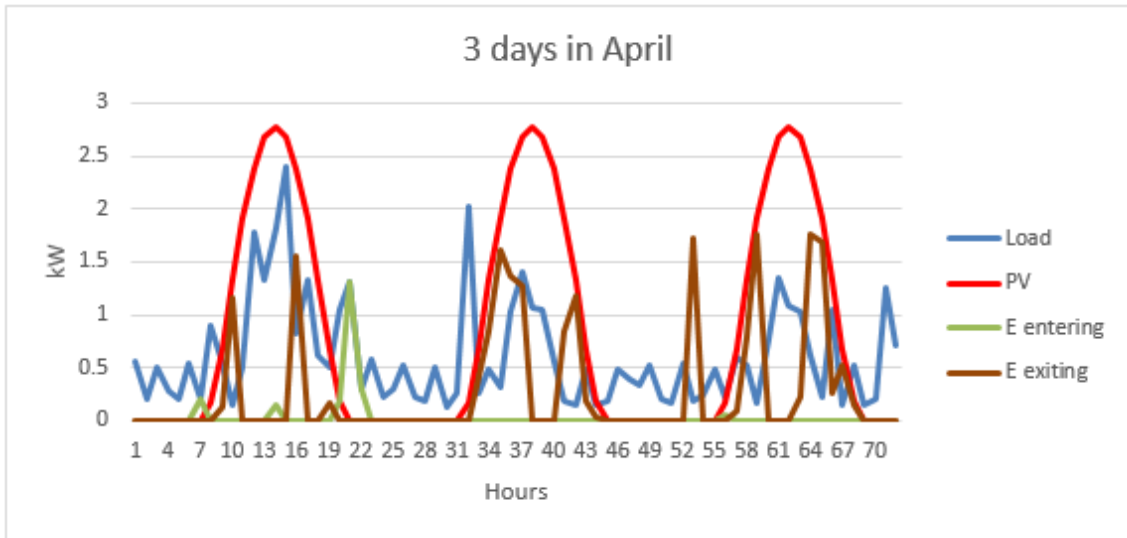


Figure 37. Sample of the output of Obama

Source: Own elaboration

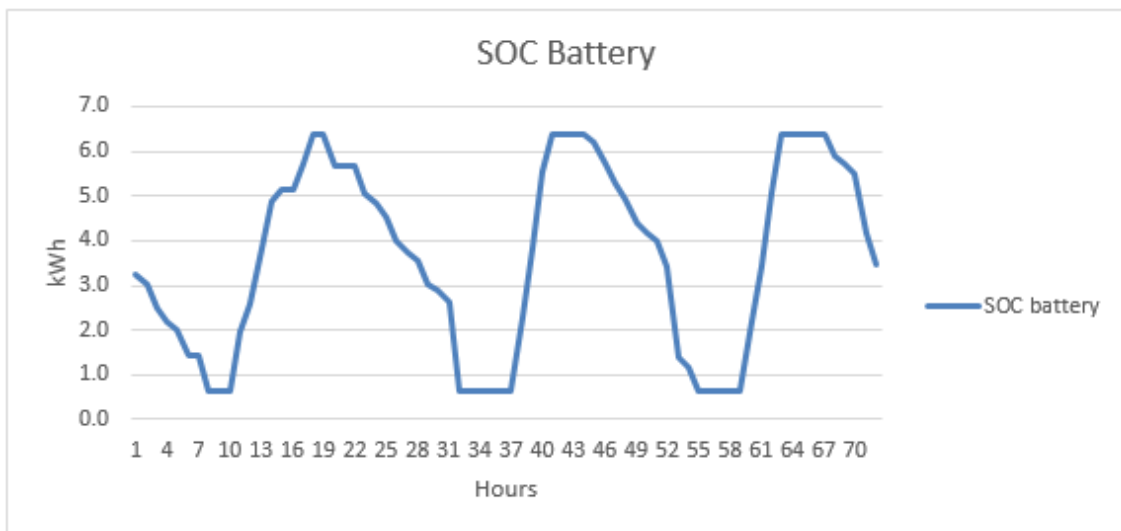


Figure 38. State of charge of the battery

Source: own elaboration

### CBA analysis:

In order to verify that the final solution chosen by OBAMA would be profitable for the Energy Community and to encourage members to invest in the technologies, a simple CBA analysis is run. The savings calculated in the reduction of the contracted power (fixed term) and in the reduction of energy consumed and the introduction of energy sold (variable term) are added up to achieve the total revenues that the self-consumption

technologies are reporting in year 2017. These revenues are extrapolated to the following 20 years that is the life assumed for the project. An inflation of 3 % is taken into account. This inflation is not only reflecting the increase of value of € but also the one of the cryptocurrency EC coin. It is believed to be a conservative value due to the big tendency existing around Blockchain and cryptocurrencies that indicate a big growth in the near future. Nevertheless, it is preferred to use a pessimistic scenario.

Figure 39 shows the cash flows calculated for the self-consumption technologies, or the equivalent of the totality of the SCP tokens.

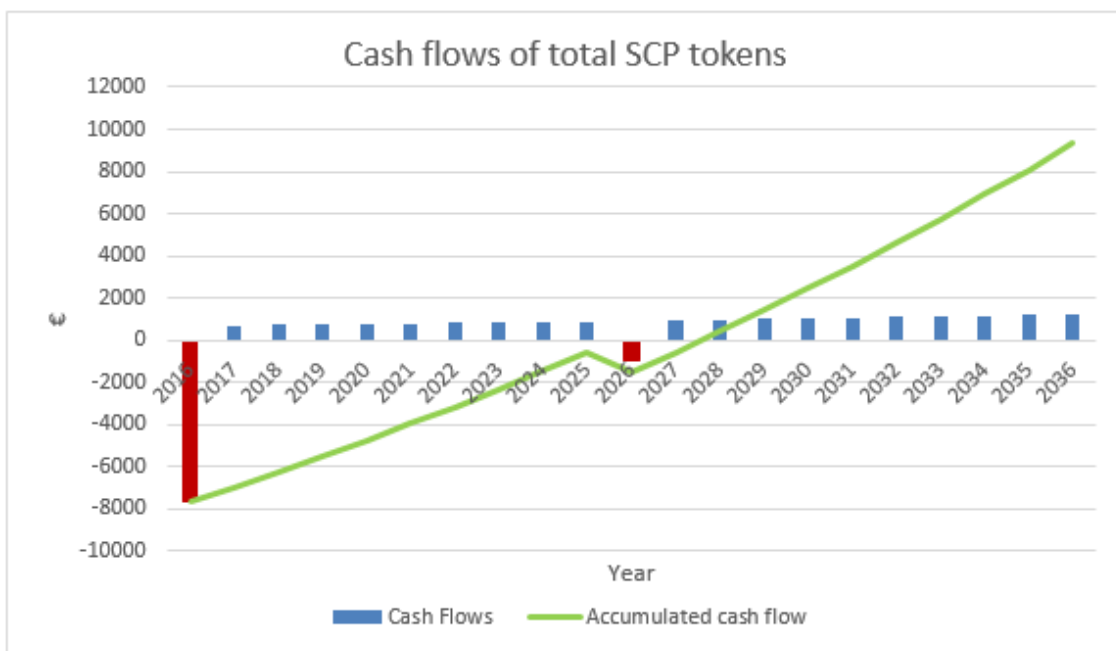


Figure 39. Cash flows of the totality of the SCP tokens of the Energy Community

Source: Own elaboration

It can be observed that after 10 years, when the life of the battery is assumed to be obsolete, a new battery of the same characteristics and dimensions is bought. It is important to outline that the price of the new battery in 10 years from now will be much cheaper looking at the projections shown in Figure 9 in 2.1. Table 11 shows some important events in the 20 years period to understand better Figure 39. Investment costs per unit of energy and power in year 0 (2016) are taken from Table 6 in Hypothesis 12.

Battery and inverter investment cost year 0:	3,700 €
PV panels investment cost year 0:	4,000 €
Total investment cost year 0	7,700€

Battery and inverter investment cost year 10 <sup>7</sup>	1840 €
Total savings = revenues year 0	702 €
Total savings = revenues year 20	1,230.4 €
<b>Break-even point</b>	<b>Year 12 = 2028</b>

*Table 11. CBA analysis calculations*

The Net Present Value (NPV) of the project has also been calculated to make the decision of buying the technologies clearer. Equation 8 shows the calculation of the NPV.

$$NPV = \sum_{t=1}^T \frac{CF_t}{(r + 1)^t} - I_0$$

*Equation 8. Calculation of NPV*

Where:

- T is equal to 20 years as commented before.
- $I_0$  is the initial investment done in year 0 equal to 7,700 €
- $CF_t$  is the cash flow generated by the savings achieved each year.
- $r$  is the discount rate. It is applied a value of 6 % that is a usual value used for batteries and PV panels. [53]

The resulting value obtained for the **NPV is 1,405.7 €**.

Since the resulting payback is 12 years, smaller than the life of the project (20 years) and the NPV is positive, the investment should be done.

Apart from the selected self-consumption technologies to be installed, since forecasts have been considered to be perfect for the whole year, OBAMA is also outputting the measurements that each of the smart meters will have in the following year. These values will be inserted in an Excel sheet and converted to a JSON file, as explained in section 7.2.5.1, to introduce them in the Dapp at the corresponding time step.

<sup>7</sup> It is calculated assuming a value of 100 € per kWh of the battery and 400 € per kW of inverter. These values are derived from Figure 9 in Chapter 2.

## 8.2 INITIALIZATION OF THE DAPP

---

To be able to start with the simulations in the Dapp, it has to be initialized. For doing so, the smart contract has to be deployed to the network and the Energy Community created. Once that is completed, a new Energy Community appears empty. Below, the initial parameters and steps taken to arrive to the start of the base case are listed.

- The Energy Community is created, and the Owner emits 250,000 EC coins. The initial price of an EC coin is set to 0.0001 ETH (ether) or the equivalent of 0.1 € (1 ETH = 1000 €).
- The Owner adds to the empty community the addresses of the 3 users, the DSO and the six smart meters with their characteristics.
- Each user as well as the DSO buys 60,000 EC coins in exchange of ETH.
- For the financing of the self-consumption technologies acquired following the optimization simulations, one PV panel of 4 kWp and one battery of 6.4 kWh, the Owner emits 770 SCP tokens with initial price of 100 EC coins each, covering in this way the investment cost of 7,700 €. As an initial forecast, it can be told to the user that 1 SCP token will cost him 100 EC coin or the equivalent of 10 € and that will report him an average benefit of 9.1 EC or 0.91 € each year.
- Users buy SCP tokens for getting rights in the generation of energy inside their community. For the base case, a decompensated distribution of tokens are done to compare different possible users' behaviors. The breakdown of SCP tokens bought by each user is shown in Table 12. There are no external investors. The totality of the SCP tokens are bought by the users

	User 1	User 2	User 3
Initial # of SCP	77	539	154
% in SCP	10 %	70 %	20 %

*Table 12. SCP tokens initially bought by each user*

Once all these steps are followed, the simulations can be started.

### 8.3 SIMULATIONS WITH THE DAPP

---

---

The Distributed Application is programmed to run in real time, so that users can keep track of the flows of energy inside the community together with the corresponding financial transactions. In addition, they can see the accumulated balance of their virtual wallets and interact with other users by exchanging tokens.

For the sake of testing and showing some results in the document, the time step at which new measurements are uploaded to the blockchain has been reduced from one hour to 20 seconds.

In the figures below, it is shown what the users see in the interface of the application for the tracking of their energy and the fluctuations in their EC coins balances. The information they are displaying corresponds to the first 24 hours of the simulation (1<sup>st</sup> of Jan 2017). Hereafter it is going to be described some insights that can be drawn out of the figures.

Figure 40 shows the flows of energy that the smart meters of the entrance of the Energy Community, the PV and the battery are measuring. It can be observed, that for the majority of the hours, the energy flow of the community is 0, which means that the loads of the users are being supplied by the internal generation of battery or PV. When the sun is rising, the battery takes advantage of the generated energy to get recharged. There is also one hour, at 13h where the Community is selling energy. In the late afternoon, the battery discharges to supply most of the demand in those hours.

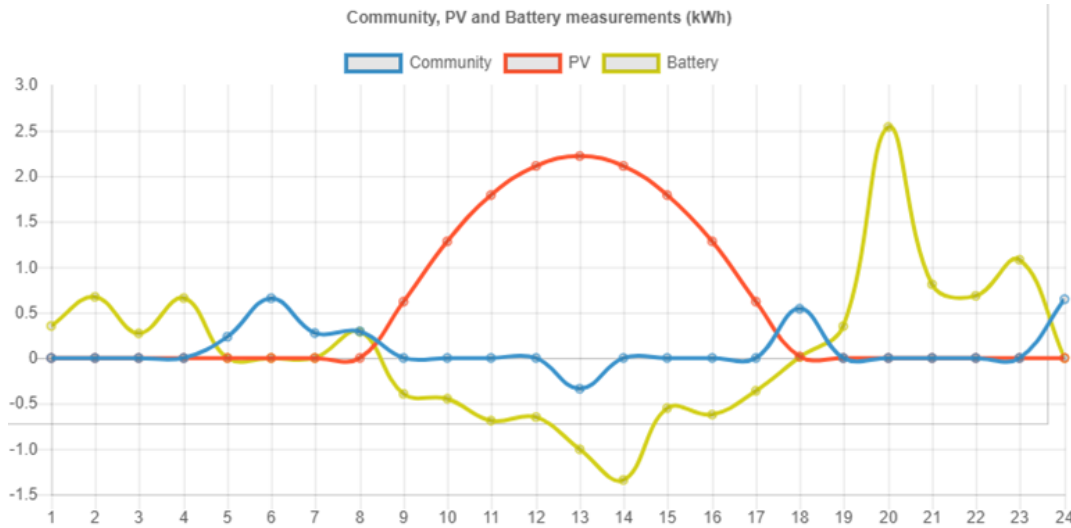


Figure 40. Community, PV and Battery smart meters measurements

Source: Own elaboration

Figure 41 displays the fluxes of EC coins that the DSO has every hour, together with the total balance it is being accumulated. It can be observed, that positive fluxes indeed correspond to hours in which the community is buying energy, while the negative one matches the exit of energy from the Community at 13h.

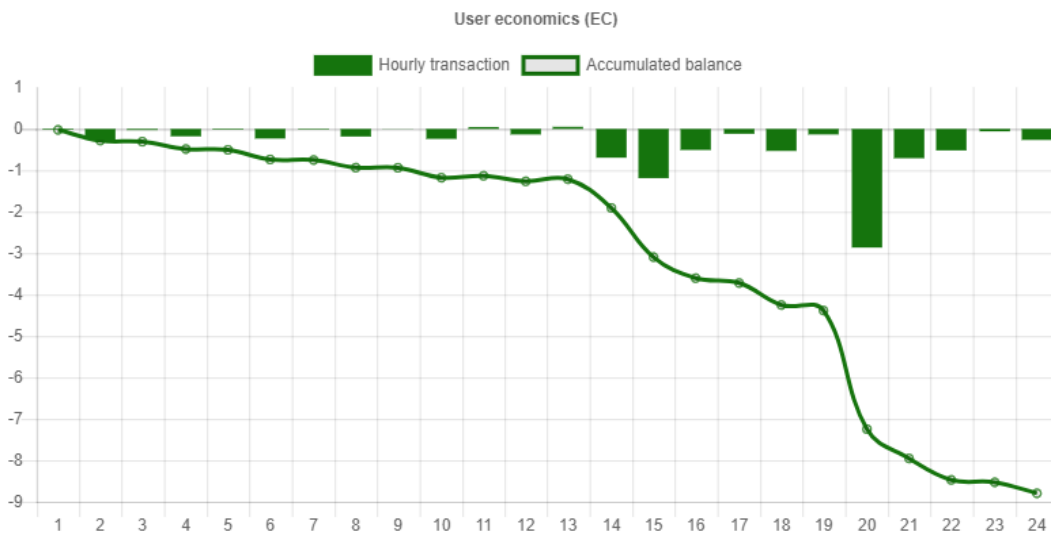


Figure 41. DSO economic transactions and accumulated EC coin balance

Source: Own elaboration

In Figure 42, it can be observed user 1 demand together with the economic flows his wallet has during the 24 hours and that are accumulated to its balance. The higher the peaks of demand, the higher the outputs of EC from his wallet are.

In the same way, Figure 43 and Figure 44 show the statistics for User 2 and User 3 respectively. It can be seen that User 2 due to the fact that has much more shares in self-consumption technologies than the others, is getting higher positive incomes of EC coins when the PV or the battery are generating energy. By the end of the first day, its balance is already equal to 8 EC coins.

To verify that there is an equilibrium of economic transactions at the end of the day and therefore the platform is properly working, the values of the balances of EC coins in each of the wallets are presented in Table 13.

User 1	-8.796 EC
User 2	7.896 EC
User 3	-0.796 EC
DSO	1.696 EC
<b>Total</b>	<b>0 EC</b>

*Table 13. EC balances at the end of day 1*

The total balance is 0 since there are no taxes applied by the Community Bank in the transactions as it was already mentioned in the hypotheses.

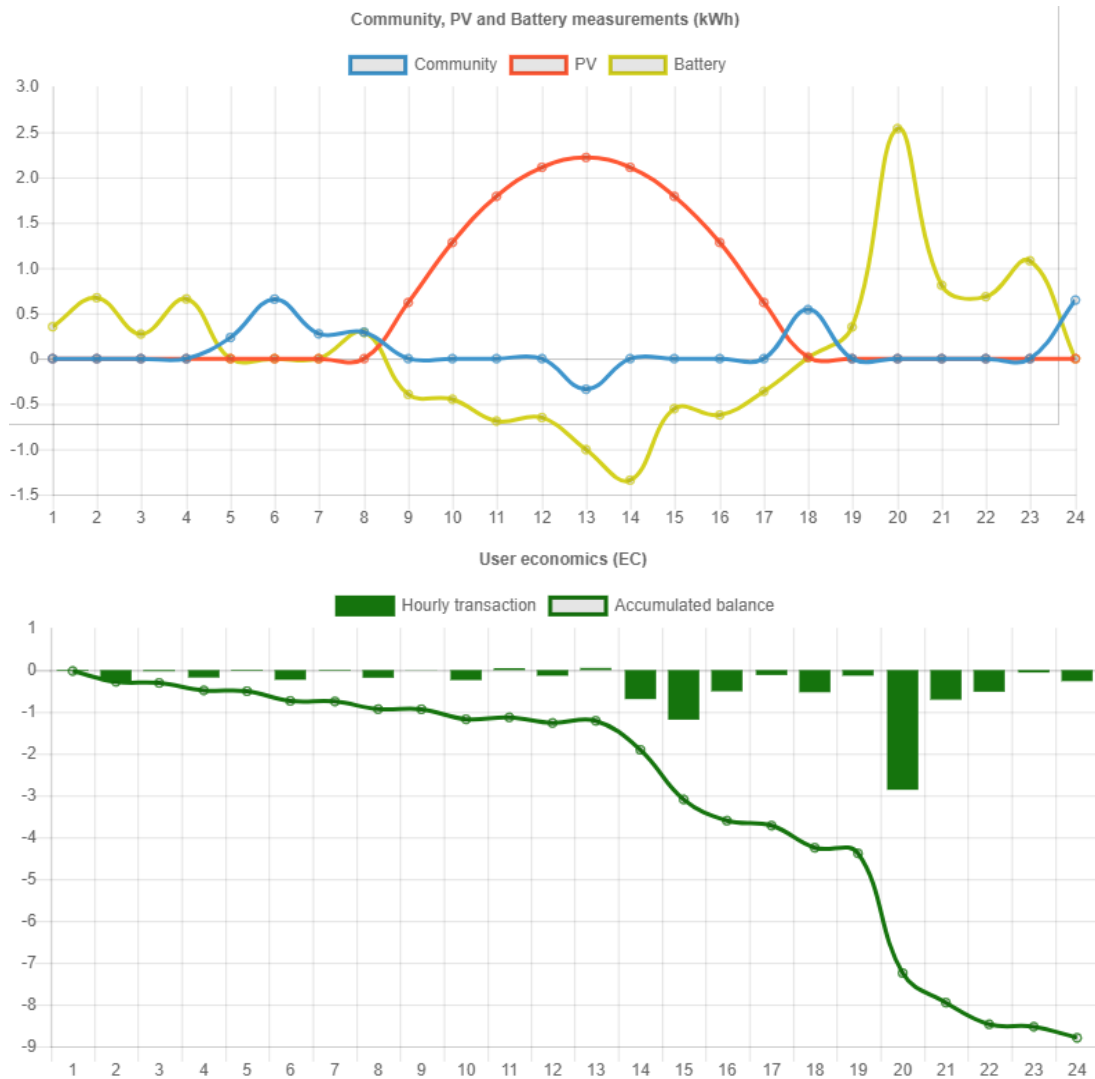


Figure 42. User 1 demand and economic flows

Source: Own elaboration



Case study, Validation of the prototype and results

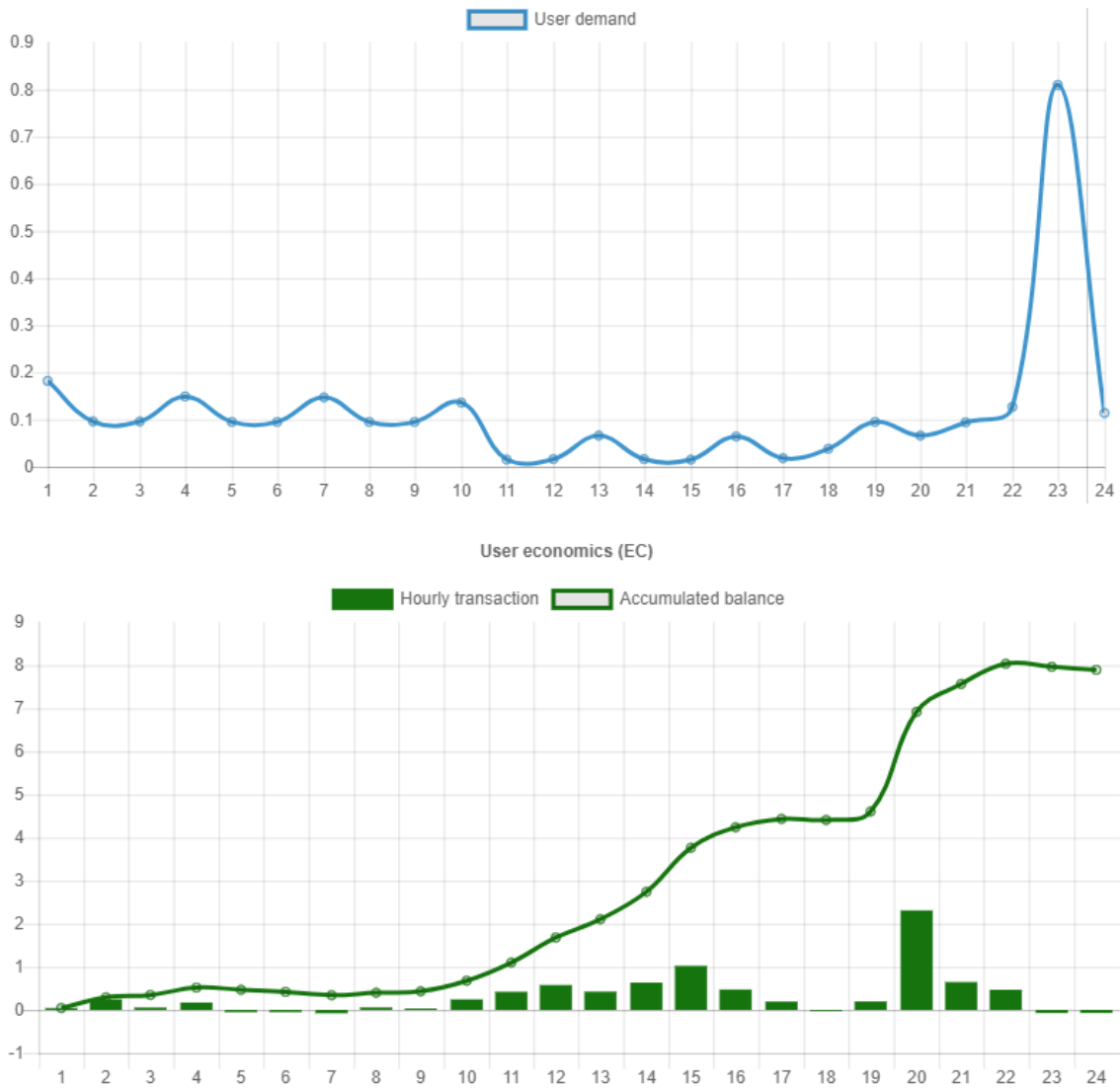


Figure 43. User 2 demand and economic flows

Source: Own elaboration

Case study, Validation of the prototype and results

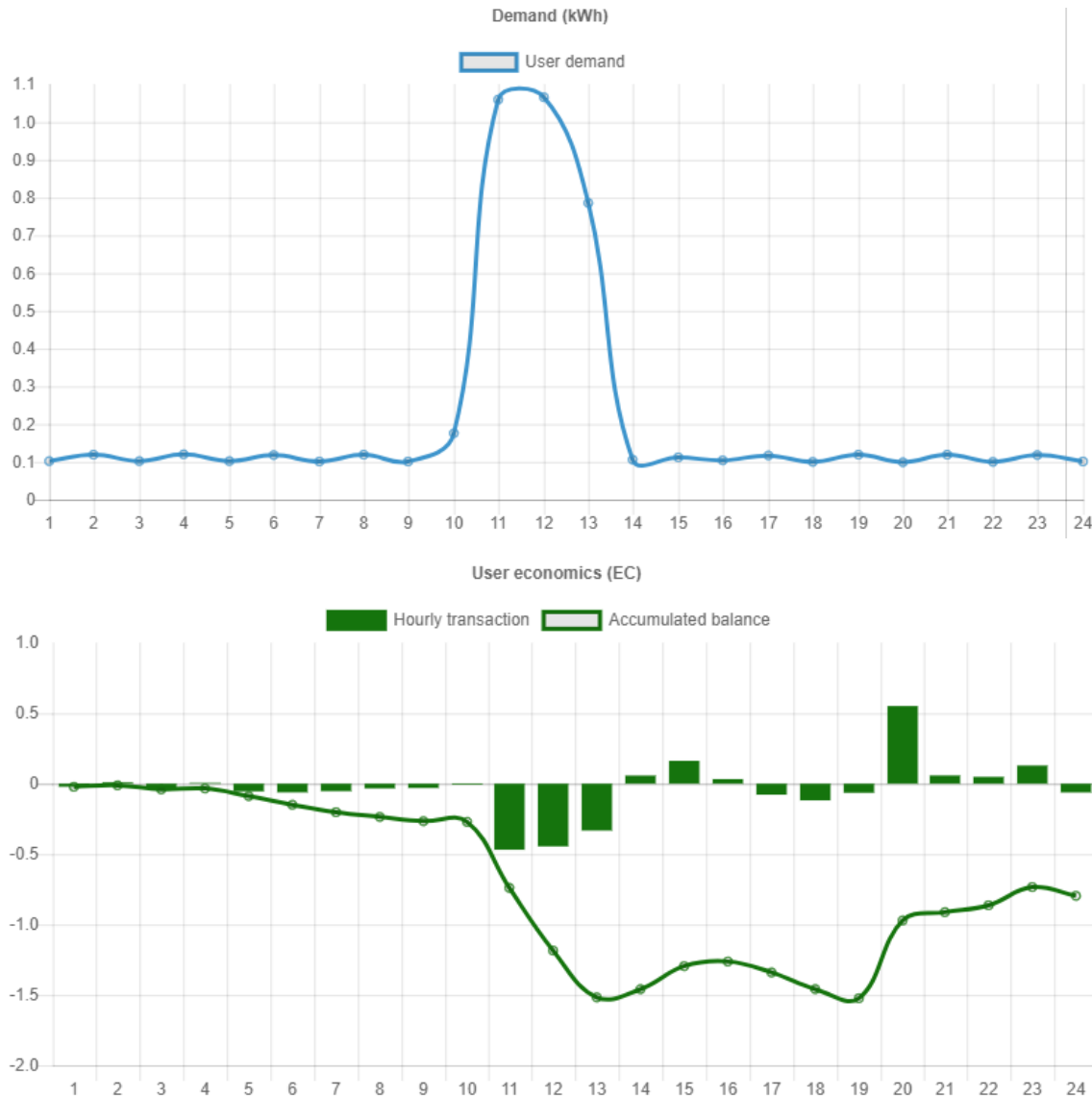


Figure 44. User 3 demand and economic flows

Source: Own elaboration

The simulations would continue day after day, until the settlement period is reached. In this case, it is set to one month. At this point of time, the total accumulated balances of each agent will be made effective and transactions between wallets with these amounts will be executed.

## 8.4 ECONOMIC ANALYSIS

Starting from the base case described, and taking as initial point the simulations shown in the previous section, in this section, a brief analysis is done to show what are the final economic flows of each user when the Dapp runs for one full year. It will also be analyzed in which months the SCP tokens report more revenues and their depreciation over the years to come.

This kind of analysis is what a user could do when being part of an Energy Community based on forecasts and the use of the App to deduct when it is more suitable for him to buy or sell shares in the self-consumption technologies inside if his Energy Community.

Table 14 shows the costs and savings for each of the users for year 2017 if they maintained the same percentages in SCP tokens as at the beginning. The table displays the resulting one year electricity bill for each user before and after the acquisition of the self-consumption technologies. From it, the savings that they brought to each user are derived.

In €	User 1 (10% SCP)			User 2 (70% SCP)			User 3 (20% SCP)		
	Old	New	Savings	Old	New	Savings	Old	New	Savings
Variable term cost	185.3	138	47.3	179.5	-151.6	331.1	447.3	352.7	94.6
Fixed term cost	29.2	6.3	22.9	217.5	57.5	160	176.7	131	45.7
<b>Total cost</b>	214.5	144.3	<b>70.2</b> <b>(33%)</b>	397	-94.2	<b>491.2</b> <b>(124%)</b>	624	483.7	<b>140.3</b> <b>(22%)</b>

*Table 14. One year electricity bill of the users inside the Energy Community*

User 1 could attain savings of 33% due to the already small total cost it had before the implementation of the PV and the battery, mostly thanks to the very low percentage in fixed term cost, with the bought of only 10% of the SCP tokens.

User 2 was the biggest investor in renewable energy and got 70% of the SCP tokens. This allowed him to pay off the entire bill and to also get an extra benefit of 94 € at the end of the year. Another conclusion that User 2 has to make is that he is paying a very high percentage of the contracted power when his overall demand is not very big, he even has a lower total energy consumption than User 1. Therefore, he needs to be more demand responsive and try to limit high peaks. By doing this, his savings will increase significantly.

User 3 is the member with the highest consumption of energy and therefore with the largest electricity bill cost. By having bought 20% of the SCP tokens, he is able to reduce his bill in a 22%.

In Figure 45, shown below, it is displayed the revenues in € that the total amount of SCP tokens is bringing to the Energy Community each of the months of the year of study. The months of summer together with January are the most profitable. Users will likely want to own SCP tokens during these months and therefore its price will most probably go up. On the other hand, November and February are the months in which less savings were achieved and thus the price of the tokens will tend to decrease during these periods.

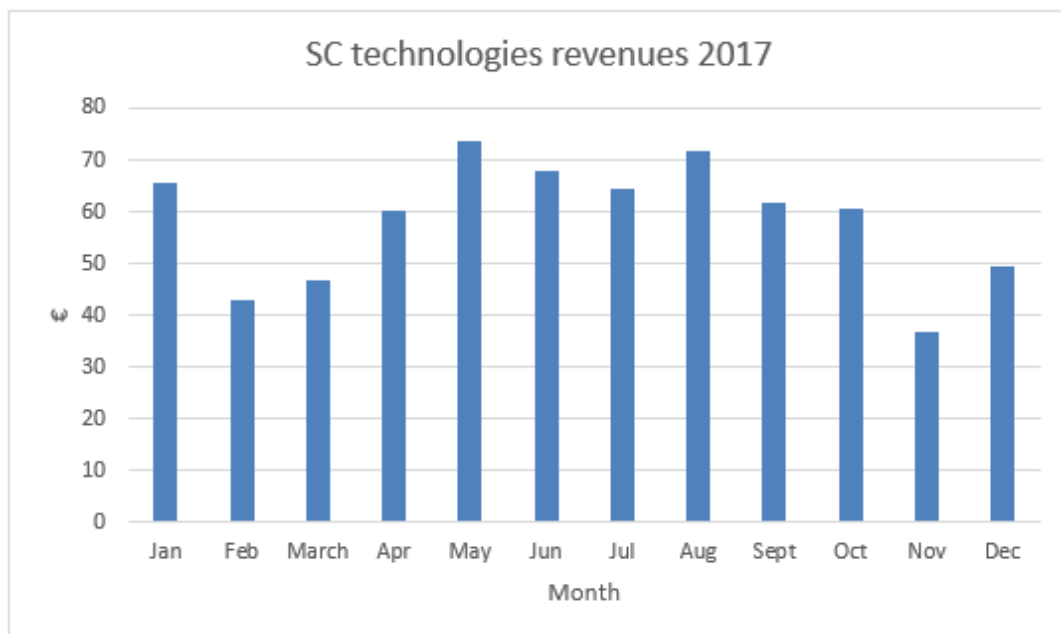


Figure 45. Revenues coming from the total amount of SCP tokens each month

Source: Own elaboration

SCP tokens value is also exposed to depreciation. With the passing of years, their value will decrease since the self-consumption technologies will be getting closer to be obsolete and therefore will be less valuable. Figure 46 shows how the value of 1 SCP token will decrease due to depreciation. A linear depreciation straight to 0 was used.

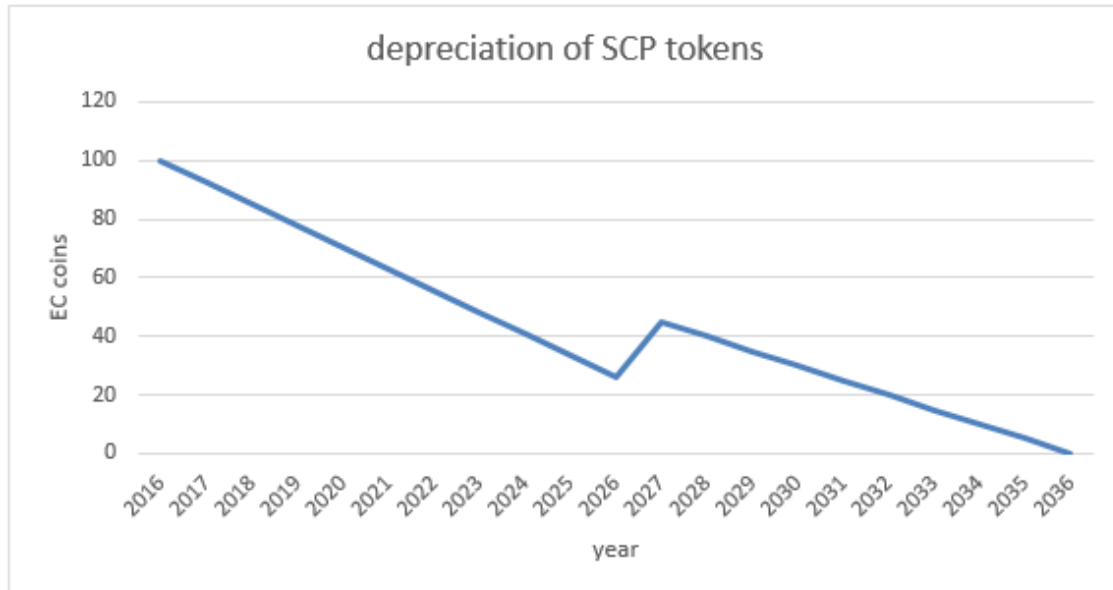


Figure 46. Depreciation of SCP tokens over the useful life of the technologies

In Figure 46 It can be observed that the initial value of 1 SCP token is 100 EC coins as commented already in the document. It is also worthy to mention the rise in value the tokens have when the second battery is bought. This makes sense since a new investment is done by the current holders of the tokens. The tokens value has to increase so that holders are able to recover the investment in case they want to sell immediately after. If this would not happen, there would be a decompensation in the internal market of the SCP tokens since no one would like to have them prior to the buying of the new battery.

Taking into account what has been described and showed in Figure 45 and Figure 46, together with a deeper analysis, would determine the price of SCP tokens at each period of time.

With the use of this application and the help of these analysis, users 1, 2 and 3 are able to invest in the renewable energy that is powering their houses in a transparent way that helps them reduce their electricity bill and improve their consumption habits. Furthermore, they will get the feeling to be socially fighting against carbon footprint and the climate change.



## Chapter 9 CONCLUSIONS

### 9.1 FINDINGS AND CONTRIBUTION TO THEORY AND PRACTICE

---

This thesis aims to enlighten the possible applications that Blockchain technology has in the power industry focusing with more depth on the local exchange of electricity. Chapter 2 describes Blockchain in an understandable way, so that the reader can grasp the main important knowledge of the technology and it can serve as the basis for future projects in the same subject. 2.1 goes through the evolution of the electricity sector in the last decades and the trends that are shaping it. These trends include: climate change with the subsequent increase of renewable energy, decentralization and digitalization show a perfect scenario for the implementation of Blockchain. There are already some researches and start-ups that are spending their money and time on creating business models that use Blockchain for electricity services, in Chapter 4 it is revised the main existing ones and their scope. It has been identified that the most popular application is the use of Blockchain for P2P exchange of energy, management of local energy communities and creation of local markets. Following this line, a Blockchain prototype application for demonstration purposes was created. Chapter 5 theoretically describes the generated prototype focusing on the suitability of Blockchain on an Energy Community and its main advantages and limitations. In addition, this chapter presents the main hypotheses and parts of the application created. Chapters 6 and 7 go over the two main parts of the prototype: 1) the optimization tool used for the operation of the energy resources and most appropriate investment in technologies and 2) the programming of the Blockchain application. Finally, Chapter 8 shows the results of the simulations run with the final prototype proving that it works properly and that it is able to automatically manage the energy community and carry the financial accounting of its users' balances. Chapter 8 also presents a simple economic analysis to understand the cash flows of each of the users during the life of the project and explains the approximated variation of prices the Self Consumption tokens would have over time.

This thesis adds value in both theory and practice in four aspects. First, it compiles the main projects that are currently working on this topic. Second, it casts light on the strengths and limitations of Blockchain technology in local energy communities and the power industry in general. Third, it shows the main steps and hints to build a simple Distributed Application using Blockchain. Fourth, it delivers a prototype that can optimize and manage an energy community and that can serve for future academic and company research as a starting point for developing a professional and commercial platform.

The application created makes congregations of consumers in energy communities more effective and less costly since they are self-managed and organized. It brings economic, social and environmental value to participant consumers. First, it reduces their electricity bill and makes it more visible so that they can understand it and improve it. Second, it gives them the chance to make smaller investments in renewable sources that are powering their houses with higher liquidity. Third, it gives them the feeling of sharing the energy they generate and forming part of a social community that fights climate change in a completely safe way on a trust-less environment. However, the author recognizes that the presented savings are strongly depended on the tariff design currently in place in Spain. Future work should analyze in detail alternative tariffs following an efficient design taking into consideration economic principles.

## 9.2 LIMITATIONS AND FUTURE WORK

---

Regarding the scope of the project, it has to be taken into account that Blockchain is a nine year old technology, still immature, and that more development needs to be made so that to reach scalability. Because of its disruptive and decentralizing nature, it clashes in many aspects with regulations and mindsets of big companies and governments. There is therefore also a need of regulatory modifications to make this technology more adaptable to the different markets.

Connected to what said before, since it is a technology in an early stage, many new projects have emerged in the last year in the form of start-ups. These projects do not offer a lot of public information and they still do not have functioning applications besides proof of concepts or pilot projects. It can be concluded that in the electricity sector, in the subject of P2P energy trading that is the more developed one, the inclusion of this technology is still at a pilot project level. In addition, because of all this, the thesis presented can be seen as a snapshot of the current situation and in a few years could become obsolete.

Regarding the prototype delivered, mainly due to the lack of resources and programming skills of the author, further development is needed to create a scalable and commercial application that can be used for a pilot project. In the next lines, it is explained the future work that can be done both in technical and operational aspects:

First, the Dapp programmed is only suitable for a community of three users. The code should be changed to make it scalable so that any number of users can be introduced.



Second, the optimization tool, OBAMA in this case, could be integrated into the Dapp so that there is no need of an intermediate step in which the results from OBAMA are collected in an Excel sheet for their subsequent introduction in the Dapp.

Third, make use of smart meter emulators that are connected to the blockchain and can sign transactions by themselves. This would make the solution more automated and closer to reality.

Fourth, prepare the Dapp to be run on a private blockchain in which different nodes can connect. This would allow for the connection of the different smart meter devices and users. In the prototype presented, only one node can be connected to the blockchain since it is all running locally in the PC.

Fifth, for more realistic results and a deeper and more complete analysis, the solution should be tested with a community with a bigger number of users and different scenarios with distinct frameworks of technology prices, environmental conditions, tariffs etc. In addition the inclusion of other technologies like heat pumps, wind turbines or electric vehicles should be taken into consideration.

Sixth, forecasting algorithms could be included in the tool. With this being done, deviations of the energy to buy or sell to the market could be taken into account getting a more realistic scenario.

Seventh, incentives for responsive demand should be included in the prototype since it would let the community have a better control of deviations and approximate to the forecasted load curve as well as control the total power peaks so that they do not surpass the contracted power.

Eighth, algorithms that define an internal market for the exchange of SCP tokens should be included to achieve a more realistic solution. Section 8.4 already presented a simple analysis of the features that should be taken into consideration to arrive to the aimed algorithms.

Ninth, study the roadblocks and easiness of implementation of this tool in the regulative frameworks of different countries.

Tenth, make a business plan of the solution with a deeper study to clarify if the proposed solution is cost-effective competitive compared to traditional models both for a retailer or cooperative that would adopt the solution as well as for participative consumers. In this study it should be analyzed the administrative and management costs and transaction fees that the Blockchain is saving to the Energy community and compare it to the investment cost of implementing the platform.

Eleventh, a deeper analysis is required to determine the compatibility of P2P energy sharing and storage systems. During the thesis it has been identified that the paybacks obtained in the results chapter are high. This is mainly due to the still high investment costs of storage systems. With the aggregation of demands inside of the community (that can be of different typologies such as households and commercial businesses) and the appropriate incentives in responsive demand, the net load curve can be significantly flattened, making batteries less effective. With P2P energy trading, batteries could become less important.

### 9.3 CONCLUSION

---

---

The electricity sector is undergoing big changes in the last decade. Global warming is fostering the introduction of renewable energy, there is being an increase of the overall demand, digitalization is increasing, consumers are becoming more active and electricity services are being introduced at a distributed level. In addition, consumers are shaping to be more interested in services than in products and the sharing economy is at its greatest splendor. Parallel to this, Blockchain technology appears. Blockchain is able to provide transparency, security, efficiency and cut costs of digital transactions based solutions and it is called to be a technology that disrupt many different businesses.

This thesis presents the main advances that have been done in the area of research and the main projects that are establishing Blockchain for P2P energy trading. It discusses the main advantages and limitations that the technology has to succeed and become scalable and finally it explains the delivered prototype consisting of a Blockchain platform that manages the internal transactions of an energy community. This platform is able to analyze the forecasts of demand curves of participants, prices curves and weather conditions and optimize the dimensions of the generation technologies to be used as well as their operation. In addition, it allows participants to invest in those technologies by buying digital tokens that can be exchanged or sold at any time. It keeps an immutable record of the measurements coming from smart meters and it couples the energy spending and generating to economic transactions that are as well automatically executed between members. Ultimately, with this platform, users can become more aware and participative in electricity services inside the community in a transparent and trustworthy way, they can invest in renewable energy in a more liquid and simple manner and they can socially and economically contribute to the energy transition towards a cleaner planet. Other applications of Blockchain to the electricity sector seem also promising to reduce transaction costs and integrate environmental-friendly solutions.

Furthermore, a necessary step to achieve an energy transition towards a decarbonized economy is to revise market designs, regulations and ultimately economic signals provided to the different stakeholders of the electricity sector.



## Chapter 10 ANNEXES

### 10.1 ANNEX A. CODE

---

#### 10.1.1 PACKAGE.JSON

---

```
{
  "name": "truffle-init-webpack",
  "version": "0.0.2",
  "description": "Frontend example using truffle v3",
  "scripts": {
    "lint": "eslint ./",
    "build": "webpack",
    "dev": "webpack-dev-server"
  },
  "author": "Douglas von Kohorn",
  "license": "MIT",
  "devDependencies": {
    "babel-cli": "^6.22.2",
    "babel-core": "^6.22.1",
    "babel-eslint": "^6.1.2",
    "babel-loader": "^6.2.10",
    "babel-plugin-transform-runtime": "^6.22.0",
    "babel-preset-env": "^1.1.8",
    "babel-preset-es2015": "^6.22.0",
    "babel-register": "^6.22.0",
    "copy-webpack-plugin": "^4.0.1",
    "css-loader": "^0.26.1",
    "eslint": "^3.14.0",
    "eslint-config-standard": "^6.0.0",
    "eslint-plugin-babel": "^4.0.0",
    "eslint-plugin-mocha": "^4.8.0",
    "eslint-plugin-promise": "^3.0.0",
    "eslint-plugin-standard": "^2.0.0",
    "excel-as-json": "^2.0.1",
    "html-webpack-plugin": "^2.28.0",
    "json-loader": "^0.5.4",
    "style-loader": "^0.13.1",
    "truffle-contract": "^1.1.11",
    "web3": "^0.20.0",
    "webpack": "^2.2.1",
    "webpack-dev-server": "^2.3.0"
  },
  "main": "truffle.js",
  "directories": {
    "test": "test"
  },
  "dependencies": {
    "file-system": "^2.2.2",
    "xlsx": "^0.12.6",
    "xlsx-to-json": "^0.3.0"
  },
  "keywords": []
}
```

## 10.1.2 TRUFFLE.JS

---

```
// Allows us to use ES6 in our migrations and tests.
require('babel-register')

module.exports = {
  networks: {
    development: {
      host: '127.0.0.1',
      port: 7545,
      network_id: '*' // Match any network id
    }
  }
}
```

### 10.1.3 WEBPACK.CONFIG.JS

```
const path = require('path');
const CopyWebpackPlugin = require('copy-webpack-plugin');

module.exports = {
  entry: './app/javascripts/app.js',
  output: {
    path: path.resolve(__dirname, 'build'),
    filename: 'app.js'
  },
  plugins: [
    // Copy our app's index.html to the build folder.
    new CopyWebpackPlugin([
      { from: './app/index.html', to: "index.html" }
    ])
  ],
  module: {
    rules: [
      {
        test: /\.css$/,
        use: [ 'style-loader', 'css-loader' ]
      }
    ],
    loaders: [
      { test: /\.json$/, use: 'json-loader' },
      {
        test: /\.js$/,
        exclude: /(node_modules|bower_components)/,
        loader: 'babel-loader',
        query: {
          presets: ['es2015'],
          plugins: ['transform-runtime']
        }
      }
    ]
  }
}
```

### 10.1.4 MIGRATIONS.SOL

---

```
pragma solidity ^0.4.17;

contract Migrations {
    address public owner;
    uint public last_completed_migration;

    modifier restricted() {
        if (msg.sender == owner) _;
    }

    function Migrations() public {
        owner = msg.sender;
    }

    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }

    function upgrade(address new_address) public restricted {
        Migrations upgraded = Migrations(new_address);
        upgraded.setCompleted(last_completed_migration);
    }
}
```

### 10.1.5 ENERGYCOMMUNITY.SOL

---

```
pragma solidity ^0.4.17;

/**
 * This contract is run by the Owner/central bank of the energy
 * community (can be a retailer or a cooperative).
 * It is prepared for the means of showing as prototype. It only works
 * with a community of 3 users and 6 Smart Meters(community, 3
 * demands(users), 1 PV, 1 Battery)
 *
 */
contract EnergyCommunity {

    address owner; // it has the function of central bank of the
    community. It issues SCP tokens and distributes payments

    event NewUserCreated(address UserAddress, uint UserId);
    event UserRemoved(uint UserId);
    event NewSmartMeterCreated(uint SMId, bool Type);
    event AllSCPTokensSold();
    event LiquidationDone(address From, address to, uint Value);
    event PowerContractedPayment(uint PowerPayment, int Percentage1,
    int Percentage2, int Percentage3);
```



```

    event NewMeasurement (uint Comm, int D1, int D2, int D3, int PV, int
    Batt);
    event NewMarketPrice (uint PoolPrice, uint AccessTariff, uint
    NetworkFee);

    struct MarketPrice{
        uint poolPrice; // price will be in €/kWh
        uint accessTariff;
        uint networkFee;
    }
    MarketPrice[] public marketPrices; //array containing all Market
    prices in time

    struct User {
        address addr;
        uint id;
        int balanceInstantaneo; // Internal balance of money that each
        user owes or is owed (<0 he owes; >0 he is owed)
        int powerPercentage; // % of power contracted
    }

    User[] public users; //array containing users

    struct SmartMeter{
        uint id;
        bool typeOfSmartMeter; // true for generation & battery (self
        consumption technologies), false for demand. There should be one for
        the community ->true
        int [] measurement; // array that holds the historic of
        measurements of this smart meter in Watios
    }

    mapping (uint => uint) public userPositionToSmartMeterId;
    mapping (uint => SmartMeter) public idToSmartMeter; //
    mapping (address => uint) public balanceSCP; //Self Consumption
    Participation tokens balance.
    mapping (address => uint) public balanceEC; //Energy Community
    coins balance. (utility token)
    mapping (address => mapping (address => uint)) public allowance;

    uint public decimals = 10**10; // in solidity we cannot use float
    numbers. We will multiply by a big number and return it to float in
    the javascript(frontend)
    uint public numberOfUsers = 0;
    uint public numberOfSmartMeters =0;
    uint public numberOfSCPTokens = 0;
    uint public numberOfSCPTokensRemaining = 0; //that have not been
    bought yet from the Owner/central bank
    uint public priceOfECCoins;
    uint public priceOfSCPTokens; //En ECCoins
    address public DSO; // Address of the Distribution System Operator
    that sells and buys energy from the community. To be filled in in the
    web app

```

```

    uint public balanceInstantaneoDSO = 0; // internal balance of the
DSO
    uint public euroToWei = 10**15; // I consider that 1 ether is
equal to 1000€. 1 ether= 10^18 wei

    uint public powerContracted = (46*decimals/10); //in kW*10^10.
initial value
    uint public powerContractedOld = 10*decimals;
    uint public initialPercentPowerUser1 = 69; //over 1000 to create
one decimal afterwards
    uint public initialPercentPowerUser2 = 513;
    uint public initialPercentPowerUser3 = 417;

    uint public maxPowerInOrOut; //to account for the maximum power
going through the smartmeter of the community
    uint public counterHours = 0;
    uint public priceOfPowerContractedPerHour = 48333000; // in
€^10/kWh. Initial value
    uint public buyingPrice;
    uint public sellingPrice;
    uint public totalDemand;
    int [] public balanceInstantaneo1Array;
    int [] public balanceInstantaneo2Array;
    int [] public balanceInstantaneo3Array;
    int [] public balanceInstantaneoDSOArray;
    int public economicTransaction1;
    int public economicTransaction2;
    int public economicTransaction3;
    int public economicTransactionDSO;
    int [] public economicTransaction1Array;
    int [] public economicTransaction2Array;
    int [] public economicTransaction3Array;
    int [] public economicTransactionDSOArray;

//CONSTRUCTOR
function EnergyCommunity (uint _initialPrice, uint _initialSupply)
public{
    owner = msg.sender;
    balanceEC[msg.sender] = _initialSupply *decimals; // EC*10e10
    priceOfECCoins = _initialPrice; //in wei

}

modifier onlyOwner() {
    require(msg.sender == owner);
    _;
}

/**

```

```

*****FROM HERE ON FUNCTIONS TO SET UP THE COMMUNITY
*****
*/
// First insert user, then its smartmeter
function addNewSmartMeter(uint _id, bool _type, uint
idUserAttachedTo) public onlyOwner{ // First insert the community SM,
then the user ones and then generation/battery. If _type is true we
dont care about idUserAttachedTo
    bool aux = false;

    idToSmartMeter[_id].id = _id;
    idToSmartMeter[_id].typeOfSmartMeter = _type;
    NewSmartMeterCreated(_id, _type);
    numberOfSmartMeters ++;

    if(_type == false){ // if we are inserting a user smart meter
        for(uint i=0; i<users.length; i++){
            if(idUserAttachedTo == users[i].id){
                userPositionToSmartMeterId[i] = _id; // we link
the user to its smart meter
                aux = true;
                break;
            }
        }
        if (aux == false) revert(); //If it is not found the smart
meter cannot be added. First add the user and then its smartmeter.
    }
}

function addNewUser(address _addr, uint _id) public onlyOwner{
    users.push(User(_addr, _id, 0, 0));
    NewUserCreated(_addr, _id);

    numberOfUsers ++;
}

function setContractedPower(uint _powerContracted, uint
_priceOfPowerContractedPerHour) public onlyOwner{
    powerContracted = _powerContracted; // in kW *decimals
    priceOfPowerContractedPerHour =
_priceOfPowerContractedPerHour; // in €*decimals
}

function removeUser (uint _id) public onlyOwner{
    uint index=10000; //random big number

    for(uint i=0; i < users.length; i++){
        if(users[i].id == _id){
            index = i;
            break;
        }
    }
    if (index == 10000){ // if id introduce was not found, exit
        revert();
    }
}

```

```

    delete users[index];
    delete userPositionToSmartMeterId[index];
    numberOfUsers--;
    UserRemoved(_id);
}

function defineDSO(address _addr) public onlyOwner{
    DSO = _addr;
}

function removeSmartMeter (uint _id) public onlyOwner{
    delete idToSmartMeter[_id];
    numberOfSmartMeters --;
}

/**
*****FROM HERE ON FUNCTIONS RELATED TO TOKENS OF THE
COMMUNITY*****
*/

function setPriceOfECCoins(uint _newPriceOfEC) public onlyOwner{
    priceOfECCoins = _newPriceOfEC; //price of EC in wei (1 ether
is 1000€) 10e15
}

function buyECCcoins (uint _numberOfCoinsToBuy) public payable{
    require(_numberOfCoinsToBuy * priceOfECCoins == msg.value); //
money paid to the contract is correct
    require((_numberOfCoinsToBuy*decimals) < balanceEC[owner]); //
Still enough tokens to sell

    balanceEC[owner] -= _numberOfCoinsToBuy*decimals;
    balanceEC[msg.sender] += _numberOfCoinsToBuy*decimals;
}

function mintSCPTokens (uint _tokensMinted, uint _pricePerToken)
public onlyOwner{
    balanceSCP[owner] += _tokensMinted;
    numberOfSCPTokens += _tokensMinted;
    numberOfSCPTokensRemaining += _tokensMinted;
    priceOfSCPTokens = _pricePerToken; //in EC*10e10
}

function buySCPTokens (uint _numberOfTokensToBuy) public{ //SCP
tokens are bought with EC coins
    require(balanceSCP[owner]>=_numberOfTokensToBuy); // Still
enough tokens to sell
    require(balanceEC[msg.sender] >= _numberOfTokensToBuy *
priceOfSCPTokens); // enough EC coins in his account

    balanceEC[msg.sender] -= _numberOfTokensToBuy *
priceOfSCPTokens;

```

```

        balanceEC[owner] += _numberOfTokensToBuy * priceOfSCPTokens;
        balanceSCP[owner] -= _numberOfTokensToBuy;
        balanceSCP[msg.sender] += _numberOfTokensToBuy;
        numberOfSCPTokensRemaining = balanceSCP[owner];
        if(balanceSCP[owner] == 0){
            AllSCPTokensSold();
        }
    }

    function burnSCPTokens (uint _tokensBurned) public onlyOwner{
        balanceSCP[owner] -= _tokensBurned;
        numberOfSCPTokens -= _tokensBurned;
        numberOfSCPTokensRemaining = balanceSCP[owner];
    }

    function burnECCoins (uint _coinsBurned) public onlyOwner{
        balanceEC[owner] -= _coinsBurned*decimals;
    }

    function buySCPTokensFromOtherUser(address _fromWho, uint
    _numberOfTokensToBuy) public{
        require(_numberOfTokensToBuy <=
        allowance[_fromWho][msg.sender]); // less than what the other user
        allowed
        require(balanceEC[msg.sender] >= _numberOfTokensToBuy *
        priceOfSCPTokens); // enough EC coins in the buyer's account

        allowance[_fromWho][msg.sender] -= _numberOfTokensToBuy;
        balanceEC[msg.sender] -= _numberOfTokensToBuy *
        priceOfSCPTokens;
        balanceEC[_fromWho] += _numberOfTokensToBuy *
        priceOfSCPTokens;
        balanceSCP[_fromWho] -= _numberOfTokensToBuy;
        balanceSCP[msg.sender] += _numberOfTokensToBuy;
    }

    function givePermission(address _permissionTo, uint _SCPQuantity)
    public{
        require(_SCPQuantity <= balanceSCP[msg.sender]);
        allowance[msg.sender][_permissionTo] = _SCPQuantity;
    }

    function withdraw() external onlyOwner {
        owner.transfer(this.balance); //withdraw money from the smart
        contract that will contain ethers that users used to pay for EC
    }

    /**
    *****FROM HERE ON GETTER
    FUNCTIONS*****
    */
    function getMyBalanceEC () public view returns(uint){
        return balanceEC[msg.sender]/decimals;
    }

```

```

}

function getMyBalanceSCP() public view returns(uint){
    return balanceSCP[msg.sender];
}
function getMybalanceInstantaneo() public view returns (int){
    uint pos = 10000;
    for(uint i=0; i< users.length; i++){
        if(users[i].addr == msg.sender){
            pos = i;
            break;
        }
    }
    if (pos == 10000){
        revert ();
    }
    return users[pos].balanceInstantaneo;
}
function getMyMeasurements() public view returns(int[]){
    uint pos = 10000;
    for(uint i=0; i< users.length; i++){
        if(users[i].addr == msg.sender){
            pos = i;
            break;
        }
    }
    if (pos == 10000){
        revert ();
    }
    return
idToSmartMeter[userPositionToSmartMeterId[pos]].measurement;
}

function getMeasurementsOfASmartMeter (uint _id) public view
returns (int[]){
    if(idToSmartMeter[_id].typeOfSmartMeter == true){ //Any user
can see measurments coming from PV, Battery and the community SM
        return idToSmartMeter[_id].measurement;
    }
    if (msg.sender == owner){
        return idToSmartMeter[_id].measurement; // the owner can
access the measurements of anyone
    }
}

function getMyInstaneousBalance(uint _idUser) public view
returns(int[]){
    if (users[_idUser-1].addr == msg.sender){
        if(_idUser == 1){
            return balanceInstantaneo1Array;
        }else if(_idUser == 2){
            return balanceInstantaneo2Array;
        }else if(_idUser == 3){
            return balanceInstantaneo3Array;
        }
    }
    }else if (msg.sender == owner){

```

```

        if(_idUser == 1){
            return balanceInstantaneo1Array;
        }else if(_idUser == 2){
            return balanceInstantaneo2Array;
        }else if(_idUser == 3){
            return balanceInstantaneo3Array;
        }
    }else{
        revert();
    }
}

function getDSOInstantaneousBalance() public view returns(int[]){
    return balanceInstantaneoDSOArray;
}

function getMyEconomicTransactions(uint _idUser) public view
returns(int[]){
    if (users[_idUser-1].addr == msg.sender){
        if(_idUser == 1){
            return economicTransaction1Array;
        }else if(_idUser == 2){
            return economicTransaction2Array;
        }else if(_idUser == 3){
            return economicTransaction3Array;
        }
    }else if (msg.sender == owner){
        if(_idUser == 1){
            return economicTransaction1Array;
        }else if(_idUser == 2){
            return economicTransaction2Array;
        }else if(_idUser == 3){
            return economicTransaction3Array;
        }
    }else{
        revert();
    }
}

function getDSOEconomicTransaction() public view returns(int[]){
    return economicTransactionDSOArray;
}

/**
***** From here on, measurement
registration functions
*****
*/

function abs(int _value) pure internal returns (int){
    if(_value >=0){
        return _value;
    }else{
        return -_value;
    }
}

```

```

    }
}

// all energy measurement are entering in kW*decimals
function newMeasurement(int _comm, int _d1, int _d2, int _d3, int
_pv, int _batt, uint _poolPrice, uint _accessTariff, uint _networkFee)
public {

    int moneyOutOrIn; // positive -> money out; negative -> money
in
    uint communitySavings;

    //register measurements and prices
    marketPrices.push(MarketPrice(_poolPrice, _accessTariff,
_networkFee));
    idToSmartMeter[0].measurement.push(_comm); //positive ->
energy entering; negative -> energy exiting
    idToSmartMeter[1].measurement.push(_d1);
    idToSmartMeter[2].measurement.push(_d2);
    idToSmartMeter[3].measurement.push(_d3);
    idToSmartMeter[4].measurement.push(_pv);
    idToSmartMeter[5].measurement.push(_batt); //positive ->
discharging; negative -> charging

    NewMeasurement(_comm, _d1, _d2, _d3, _pv, _batt);
    NewMarketPrice(_poolPrice, _accessTariff, _networkFee);

    //Calculations of financials derived from measurements and
prices
    totalDemand = uint(_d1 + _d2 + _d3);
    buyingPrice = (_poolPrice + _accessTariff + _networkFee); //
en €*10^10/kW
    sellingPrice = (_poolPrice - _networkFee);

    if(_comm > 0){ //energy entering the comm
        moneyOutOrIn = (_comm * int(buyingPrice))/int(decimals);
//We erase one of the decimals(10^10) since we are multiplying 2
quantities that have decimals
        if(totalDemand < uint(_comm)){
            communitySavings = 0;
        }else{ // there is being some generation inside the
community
            communitySavings = (totalDemand - uint(_comm)) *
buyingPrice/decimals;
        }
    }else if(_comm == 0){
        moneyOutOrIn = 0;
        communitySavings = (totalDemand * buyingPrice)/decimals;
    }else{//energy exiting
        moneyOutOrIn = (_comm * int(sellingPrice))/int(decimals);
    }
}
}

```



```

    communitySavings = (totalDemand * buyingPrice)/decimals +
    uint(-moneyOutOrIn); //minus before moneyOutOrIn to have a positive
    number

  }

  //Update internal balances of all users and DSO in EC
  economicTransaction1 = (_d1 * int(buyingPrice)/int(decimals)-
  int(communitySavings *
  balanceSCP[users[0].addr]/numberOfSCPTokens))*int(euroToWei)/int(price
  OfECCoins); // save in EC *10^10
  economicTransaction2 = (_d2 * int(buyingPrice)/int(decimals)-
  int(communitySavings *
  balanceSCP[users[1].addr]/numberOfSCPTokens))*int(euroToWei)/int(price
  OfECCoins);
  economicTransaction3 = (_d3 * int(buyingPrice)/int(decimals)-
  int(communitySavings *
  balanceSCP[users[2].addr]/numberOfSCPTokens))*int(euroToWei)/int(price
  OfECCoins);
  economicTransactionDSO =
  moneyOutOrIn*int(euroToWei)/int(priceOfECCoins);
  users[0].balanceInstantaneo += economicTransaction1;
  users[1].balanceInstantaneo += economicTransaction2;
  users[2].balanceInstantaneo += economicTransaction3;
  balanceInstantaneoDSO += economicTransactionDSO;
  // update to arrays for storage and post plotting
  economicTransaction1Array.push(economicTransaction1);
  economicTransaction2Array.push(economicTransaction2);
  economicTransaction3Array.push(economicTransaction3);
  economicTransactionDSOArray.push(economicTransactionDSO);
  balanceInstantaneo1Array.push(users[0].balanceInstantaneo);
  balanceInstantaneo2Array.push(users[1].balanceInstantaneo);
  balanceInstantaneo3Array.push(users[2].balanceInstantaneo);
  balanceInstantaneoDSOArray.push(balanceInstantaneoDSO);

  //add one to the number of hours for after liquidation of
  contracted powrers
  counterHours ++;
}

/**
*****LIQUIDATIONS*****
**/

function transactionECCoins (address _from, address _to, uint
_value) internal{
  require(balanceEC[_from] >= (_value));
  balanceEC[_from] -= _value;
  balanceEC[_to] += _value;
  LiquidationDone(_from, _to, _value);
}

```

```

function liquidationOfTransactions () public {
    uint powerContractedFee;
    if(balanceInstantaneoDSO>0){

        transactionECCoins(owner,DSO,
uint(balanceInstantaneoDSO)); //We pass value in EC*decimals
    }else{

        transactionECCoins(DSO,owner, uint(-
balanceInstantaneoDSO));
    }
    balanceInstantaneoDSO = 0; //Reinitialize the count for
balanceinsnataneo
    for(uint i=0; i<users.length; i++){

        if(users[i].balanceInstantaneo>0){
            transactionECCoins(users[i].addr,owner,
uint(users[i].balanceInstantaneo));
        }else{
            transactionECCoins(owner,users[i].addr, uint(-
users[i].balanceInstantaneo));
        }
        users[i].balanceInstantaneo = 0; //Reinitialize count for
user internal balance
    }

    //Power payment from owner/central bank to DSO for the
contracted power of the community
    balanceEC[owner] -=
powerContracted*priceOfPowerContractedPerHour/decimals *counterHours*
euroToWei/priceOfECCoins;
    balanceEC[DSO] +=
powerContracted*priceOfPowerContractedPerHour/decimals *counterHours*
euroToWei/priceOfECCoins;

    //each user pays depending on their contribution percentage to
the Owner:
    users[0].powerPercentage =
(int(initialPercentPowerUser1*powerContractedOld/1000)-
int((powerContractedOld -
powerContracted)*balanceSCP[users[0].addr]/numberOfSCPTokens))*1000/in
t(powerContracted);
    users[1].powerPercentage =
(int(initialPercentPowerUser2*powerContractedOld/1000)-
int((powerContractedOld -
powerContracted)*balanceSCP[users[1].addr]/numberOfSCPTokens))*1000/in
t(powerContracted);
    users[2].powerPercentage =
(int(initialPercentPowerUser3*powerContractedOld/1000)-
int((powerContractedOld -
powerContracted)*balanceSCP[users[2].addr]/numberOfSCPTokens))*1000/in
t(powerContracted);

    for(uint j=0; j<users.length; j++){
        if(users[j].powerPercentage > 0){ //if they pay

```

```

        powerContractedFee =
powerContracted*priceOfPowerContractedPerHour/decimals *counterHours*
uint(users[j].powerPercentage)/1000 * euroToWei/priceOfECCoins;
        balanceEC[users[j].addr] -= powerContractedFee;
        balanceEC[owner] += powerContractedFee;
    }else{ // if negative they get an incentive for it
        powerContractedFee =
powerContracted*priceOfPowerContractedPerHour/decimals *counterHours*
uint(-users[j].powerPercentage)/1000 * euroToWei/priceOfECCoins;
        balanceEC[owner] -= powerContractedFee;
        balanceEC[users[j].addr] += powerContractedFee;
    }
}

PowerContractedPayment (powerContracted*priceOfPowerContractedPerHour/d
ecimals *counterHours* euroToWei/priceOfECCoins,
users[0].powerPercentage, users[1].powerPercentage,
users[2].powerPercentage);
    counterHours = 0; //initialize counter of hours for max power
measurements
    maxPowerInOrOut =0;
}
}

```

### 10.1.6 1\_INITIAL\_MIGRATION.JS

```

var Migrations = artifacts.require("./Migrations.sol");

module.exports = function(deployer) {
    deployer.deploy(Migrations);
};

```

### 10.1.7 2\_DEPLOY\_CONTRACTS.JS

```

var EnergyCommunity = artifacts.require("./energyCommunity.sol")

module.exports = function(deployer) {
    deployer.deploy(EnergyCommunity, web3.toWei(0.0001,'ether'),250000);
};

```

### 10.1.8 APP.JS

```

// Import the page's CSS. Webpack will know what to do with it.
import "../stylesheets/app.css";

// Import libraries we need.

```

```
import { default as Web3} from 'web3';
import { default as contract } from 'truffle-contract'
import Chart from 'chart.js';

// Import our contract artifacts and turn them into usable
abstractions.

import energycommunity_artifacts from
'../../build/contracts/EnergyCommunity.json'

// EnergyCommunity is our usable abstraction, which we'll use through
the code below.

var EnergyCommunity = contract(energycommunity_artifacts);

var euroToWei = Math.pow(10,15);
var decimals = Math.pow(10,10);
var accounts;
var account;
var lengthOfUsers;
var ownerOrUser = 0;
var hourCounter = 0;
var hourNonReset = 0;
var myVar;

var timeStep = 20000; // in milliseconds

window.App = {

start: function() {
  var self = this;

  // Bootstrap the EnergyCommunity abstraction for Use.
  EnergyCommunity.setProvider(web3.currentProvider);

  // Get the initial account balance so it can be displayed.
  web3.eth.getAccounts(function(err, accs) {
    if (err != null) {
      alert("There was an error fetching your accounts.");
      return;
    }
    alert(accs.length);
    if (accs.length == 0) {
      alert("Couldn't get any accounts! Make sure your Ethereum
client is configured correctly.");
      return;
    }
  })
}
```

```

accounts = accs;
account = accounts[0];

self.refreshBalanceEC ();
self.refreshBalanceSCP ();
self.refreshNumberOfSCPTokens ();
self.refreshPrices ();
self.refreshNumberOfUsers ();
self.refreshNumberOfSmartMeters ();
self.refreshBalanceInstantaneo ();
self.refreshBalanceInstantaneoDSO ();
self.refreshPowerContracted ();
self.refreshHour ();
});
},

startSimulation:function () {
myVar = setInterval (App.excelSheet, timeStep);
},

stopSimulation:function () {
window.clearInterval (myVar);
},

resetSimulation:function () {
window.clearInterval (myVar);
var hourCounter = localStorage.getItem ("hourCounter");
var hourNonReset = localStorage.getItem ("hourNonReset");

hourNonReset = Number (hourNonReset) + Number (hourCounter);
hourCounter = 0;

localStorage.setItem ("hourCounter", hourCounter);
localStorage.setItem ("hourNonReset", hourNonReset);

},

excelSheet: function () {
var hourCounter = localStorage.getItem ("hourCounter");
var jsonfile=JSON.parse ('PASTE HERE data.json');
console.log (jsonfile [hourCounter]);
console.log (jsonfile [hourCounter].comm);

var self = this;

var meta;
EnergyCommunity.deployed ().then (function (instance) {
meta = instance;

```

```

        return meta.newMeasurement (jsonfile[hourCounter].comm*decimals,
jsonfile[hourCounter].d1*decimals, jsonfile[hourCounter].d2*decimals,
jsonfile[hourCounter].d3*decimals, jsonfile[hourCounter].pv*decimals,
jsonfile[hourCounter].bat*decimals,
jsonfile[hourCounter].omiePrice*decimals,
jsonfile[hourCounter].accessTariff*decimals,
jsonfile[hourCounter].networkFee*decimals, {from: account});

    }).catch(function(e) {
        console.log(e);
        //self.setStatus("Error getting measurmentes; see log.");
    });
    localStorage.setItem("hourCounter", hourCounter);

    setTimeout(App.addHour, 5000);

},

refreshHour: function(){
    var hourCounter = localStorage.getItem("hourCounter");
    var aux1 = document.getElementById("currentHour");
    aux1.innerHTML = hourCounter;
    localStorage.setItem("hourCounter", hourCounter);
},

addHour: function(){
    var self = this;
    var hourCounter = localStorage.getItem("hourCounter");
    hourCounter++;
    var aux1 = document.getElementById("currentHour");
    aux1.innerHTML = hourCounter;
    localStorage.setItem("hourCounter", hourCounter);
},

liquidationOfTransactions: function(){
    var self = this;

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.liquidationOfTransactions({from: account,
gas:3000000});
    }).then(function() {
        self.refreshBalanceInstantaneo();
        self.refreshBalanceInstantaneoDSO();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error in liquidation; see log.");
    });
},

openTab: function(evt, tab) {
    // Declare all variables
    var self = this;
    var i, tabcontent, tablinks;

```

```

// Get all elements with class="tabcontent" and hide them
tabcontent = document.getElementsByClassName("tabcontent");
for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
}

// Get all elements with class="tablinks" and remove the class
"active"
tablinks = document.getElementsByClassName("tablinks");
for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace("
active", "");
}

// Show the current tab, and add an "active" class to the button
that opened the tab
document.getElementById(tab).style.display = "block";
evt.currentTarget.className += " active";
if(tab=="Owner"){
    ownerOrUser = 0;
    self.refreshBalanceEC();
    self.refreshBalanceSCP();
}
if(tab=="User"){
    ownerOrUser = 1;
    self.refreshBalanceEC();
    self.refreshBalanceSCP();
}
if(tab=="Events"){
    self.events();
}
},
plot: function(){
    var self = this;
    var hourCounter = localStorage.getItem("hourCounter");
    var hourNonReset = localStorage.getItem("hourNonReset");

    var id = parseInt(document.getElementById("idUSERToShow").value);
    var fromh = parseInt(document.getElementById("fromh").value);
    var toh = parseInt(document.getElementById("toh").value);

    var dataDemandUser = [];
    var dataPV = [];
    var dataComm = [];
    var dataBatt = [];
    var dataInstBal =[];
    var dataInstBalDSO = [];
    var dataEconomics =[];
    var dataEconomicsDSO = [];
    var time=[];

    //feel array time with hours of simulations
    for(var i = fromh; i < toh+1; i ++){
        time.push(i);
    }

```

```

    }
    //Get measurements from smartmeters from smart contract for their
    plots
    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.getMeasurementsOfASmartMeter(id, {from: account});
    }).then(function(value) {
        var aux2 = value.valueOf();
        for(var i = fromh-1+Number(hourNonReset); i <
toh+Number(hourNonReset); i ++){
            var auxil = aux2[i]/decimals;
            dataDemandUser.push(auxil);
        }

    var ctx = document.getElementById("demand");
    var myChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: time,
            datasets: [{
                data: dataDemandUser,
                label: "User demand",
                borderColor: "#3e95cd",
                fill: false
            }
        ]
    },
    options: {
        title: {
            display: true,
            text: 'Demand (kWh)'
        }
    }
    });
});

EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.getMeasurementsOfASmartMeter(0, {from: account});
}).then(function(value) {
    var aux2 = value.valueOf();
    for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i ++){
        var auxil = aux2[i]/decimals;
        dataComm.push(auxil);
    }

}).then(EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.getMeasurementsOfASmartMeter(4, {from: account});
}).then(function(value) {
    var aux2 = value.valueOf();
    for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i ++){
        dataPV.push(aux2[i]/decimals);
    }
}

```



```

    }
  })).then(EnergyCommunity.deployed()).then(function(instance) {
    meta = instance;
    return meta.getMeasurementsOfASmartMeter(5, {from: account});
  }).then(function(value) {
    var aux2 = value.valueOf();
    for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i++){
      dataBatt.push(aux2[i]/decimals);
    }
    var ctx = document.getElementById("commPVBat");
    var myChart = new Chart(ctx, {
      type: 'line',
      data: {
        labels: time,
        datasets: [{
          data: dataComm,
          label: "Community",
          borderColor: "#3e95cd",
          fill: false
        }, {
          data: dataPV,
          label: "PV",
          borderColor: "#FF5733",
          fill: false
        }, {
          data: dataBatt,
          label: "Battery",
          borderColor: "#D4D11D",
          fill: false
        }
      ]
    },
    options: {
      title: {
        display: true,
        text: 'Community, PV and Battery measurements (kWh)'
      }
    }
  });
}));

//Get data from economic transactions arrays
EnergyCommunity.deployed().then(function(instance) {
  meta = instance;
  return meta.getMyEconomicTransactions(id, {from: account});
}).then(function(value) {
  var aux2 = value.valueOf();
  for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i++){
    var auxil = aux2[i]/decimals*(-1);
    dataEconomics.push(auxil);
  }

}).then(EnergyCommunity.deployed()).then(function(instance) {
  meta = instance;

```

```

    return meta.getMyInstaneousBalance(id, {from: account});
  }).then(function(value) {
    var aux2 = value.valueOf();
    for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i ++){
      dataInstBal.push(aux2[i]/decimals*(-1));
    }
    var ctx = document.getElementById("economictransactions");
    var myChart = new Chart(ctx, {
      type: 'bar',
      data: {
        labels: time,
        datasets: [{
          data: dataEconomics,
          label: "Hourly transaction",
          borderColor: "#15740D",
          backgroundColor: "#15740D",
          fill: true
        }, {
          data: dataInstBal,
          label: "Accumulated balance",
          borderColor: "#15740D",
          fill: false,
          type: 'line'
        }
      ]
    },
    options: {
      title: {
        display: true,
        text: 'User economics (EC)'
      }
    }
  });
});

//Get data from instantaneous balance arrays
EnergyCommunity.deployed().then(function(instance) {
  meta = instance;
  return meta.getDSOEconomicTransaction({from: account});
}).then(function(value) {
  var aux2 = value.valueOf();
  for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i ++){
    var auxil = aux2[i]/decimals;
    dataEconomicsDSO.push(auxil);
  }

}).then(EnergyCommunity.deployed().then(function(instance) {
  meta = instance;
  return meta.getDSOInstataneousBalance({from: account});
}).then(function(value) {
  var aux2 = value.valueOf();
  for(var i = (fromh-1+Number(hourNonReset)); i <
(toh+Number(hourNonReset)); i ++){
    dataInstBalDSO.push(aux2[i]/decimals);
  }

```

```

    }
    var ctx = document.getElementById("instBal");
    var myChart = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: time,
            datasets: [{
                data: dataEconomicsDSO,
                label: "hourly transaction",
                borderColor: "#7A1288",
                backgroundColor: "#7A1288",
                fill: false
            }, {
                data: dataInstBalDSO,
                label: "Accumulated balance",
                borderColor: "#7A1288",
                fill: false,
                type: 'line'
            }
        ]
    },
    options: {
        title: {
            display: true,
            text: 'DSO economics (EC)'
        }
    }
    });
});

},

setStatus: function(message) {
    var status = document.getElementById("status");
    status.innerHTML = message;
},

refreshNumberOfUsers: function(){
    var self = this;

    var metal;
    EnergyCommunity.deployed().then(function(instance) {
        metal=instance;
        return metal.numberOfWorkers.call();
    }).then(function(value) {
        lengthOfWorkers = value.valueOf();
        var balance_element = document.getElementById("usersLength");
        balance_element.innerHTML = lengthOfWorkers;
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error displaying number of users; see log.");
    });
},

refreshBalanceInstantaneoDSO: function(){

```

```

var self = this;

var metal;
EnergyCommunity.deployed().then(function(instance) {
    metal=instance;
    return metal.balanceInstantaneoDSO.call();
}).then(function(value) {
    var aux = value.valueOf();
    var balance_element =
document.getElementById("balanceInstantaneoDSO");
    balance_element.innerHTML = aux/decimals;
}).catch(function(e) {
    console.log(e);
    self.setStatus("Error displaying balance instantaneo del DSO;
see log.");
});
},

refreshBalanceInstantaneo: function() {
    var self = this;

    var metal;
    EnergyCommunity.deployed().then(function(instance) {
        metal=instance;
        return metal.users(0);
    }).then(function(value) {
        var aux = value.valueOf();
        var balance_element =
document.getElementById("balanceInstantaneo1");
        balance_element.innerHTML = aux[2]/decimals*(-1);
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error displaying balance instantaneo; see
log.");
    });

    var meta2;
    EnergyCommunity.deployed().then(function(instance) {
        meta2=instance;
        return meta2.users(1);
    }).then(function(value) {
        var aux = value.valueOf();
        var balance_element =
document.getElementById("balanceInstantaneo2");
        balance_element.innerHTML = aux[2]/decimals*(-1);
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error displaying balance instantaneo; see
log.");
    });

    var meta3;
    EnergyCommunity.deployed().then(function(instance) {

```

```

        meta3=instance;
        return meta3.users(2);
    }).then(function(value) {
        var aux = value.valueOf();
        var balance_element =
document.getElementById("balanceInstantaneo3");
        balance_element.innerHTML = aux[2]/decimals*(-1);
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error displaying balance instantaneo; see
log.");
    });
},

refreshNumberOfSmartMeters: function(){
    var self = this;

    var meta1;
    EnergyCommunity.deployed().then(function(instance){
        meta1=instance;
        return meta1.numberofSmartMeters.call();
    }).then(function(value) {
        var numberOfSM = value.valueOf();
        var aux = document.getElementById("numberOfSM");
        aux.innerHTML = numberOfSM;
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error displaying number of users; see log.");
    });
},

userChange: function(){
    var self = this;
    var x = document.getElementById("userList").value;
    document.getElementById("user").innerHTML = x;
},

refreshBalanceEC: function() {
    var self = this;

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.getMyBalanceEC.call({from: account});
    }).then(function(value) {
        if(ownerOrUser == 0){
            var balance_element =
document.getElementById("balanceECOwner");
            balance_element.innerHTML = value.valueOf();
        }else{
            var balance_element =
document.getElementById("balanceECUser");
            balance_element.innerHTML = value.valueOf();
        }
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error getting balance EC; see log.");
    });
}

```

```

    });
  },

  refreshBalanceSCP: function() {
    var self = this;

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
      meta = instance;
      return meta.getMyBalanceSCP.call({from: account});
    }).then(function(value) {
      if(ownerOrUser == 0){
        var balance_element =
document.getElementById("balanceSCPowner");
        balance_element.innerHTML = value.valueOf();
      }else{
        var balance_element =
document.getElementById("balanceSCPUser");
        balance_element.innerHTML = value.valueOf();
      }

    }).catch(function(e) {
      console.log(e);
      self.setStatus("Error getting balance SCP; see log.");
    });
  },

  refreshPrices: function(){
    var self = this;

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
      meta=instance;
      return meta.priceOfECCoins.call({from: account});
    }).then(function(value) {
      var aux1 = document.getElementById("priceOfECCoins");
      aux1.innerHTML = (value.valueOf()/euroToWei);
    });
    EnergyCommunity.deployed().then(function(instance) {
      meta=instance;
      return meta.priceOfSCPTokens.call({from: account});
    }).then(function(value) {
      var aux2 = document.getElementById("priceOfSCPTokens");
      aux2.innerHTML = (value.valueOf()/decimals);
    }).catch(function(e) {
      console.log(e);
      self.setStatus("Error displaying prices; see log.");
    });
  },

  refreshPowerContracted: function(){
    var self = this;

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
      meta=instance;

```

```

    return meta.powerContracted.call({from: account});
  }).then(function(value) {
    var aux1 = document.getElementById("ContPowerComm");
    aux1.innerHTML = (value.valueOf()/decimals);
  });

EnergyCommunity.deployed().then(function(instance) {
  meta=instance;
  return meta.priceOfPowerContractedPerHour.call({from: account});
}).then(function(value) {
  var aux2 = document.getElementById("ContPowerPrice");
  aux2.innerHTML = (value.valueOf()/decimals*24);
});

EnergyCommunity.deployed().then(function(instance) {
  meta=instance;
  return meta.maxPowerInOrOut.call({from: account});
}).then(function(value) {
  var aux3 = document.getElementById("maxPowerInOrOut");
  aux3.innerHTML = (value.valueOf()/decimals);
});
},

refreshNumberOfSCPTokens: function() {
  var self = this;

  var meta;
  EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.numberOfSCPTokens.call({from: account});
  }).then(function(value) {
    var aux1 = document.getElementById("numberOfSCPTokens");
    aux1.innerHTML = value.valueOf();
  });
  EnergyCommunity.deployed().then(function(instance) {
    meta=instance;
    return meta.numberOfSCPTokensRemaining.call({from: account});
  }).then(function(value) {
    var aux2 =
document.getElementById("numberOfSCPTokensRemaining");
    aux2.innerHTML = value.valueOf();
  }).catch(function(e) {
    console.log(e);
    self.setStatus("Error displaying number of SCP tokens; see
log.");
  });
},

mintSCPTokens: function() {
  var self = this;

  var amount = parseInt(document.getElementById("amount").value);
  var price = document.getElementById("priceSCP").value * decimals;

  this.setStatus("Initiating transaction... (please wait)");

```

```

var meta;
EnergyCommunity.deployed().then(function(instance) {
  meta = instance;
  return meta.mintSCPTokens(amount, price, {from: account});
}).then(function() {
  self.setStatus("Transaction complete!");
  self.refreshBalanceSCP();
  self.refreshNumberOfSCPTokens();
  self.refreshPrices();
}).catch(function(e) {
  console.log(e);
  self.setStatus("Error minting SCP tokens; see log.");
});
},

getMeasurementsOfASmartMeter: function() {
  var self = this;

  var id = parseInt(document.getElementById("idSMTToShow").value);
  var aux2 = document.getElementById("idOfSm");
  aux2.innerHTML = id;
  this.setStatus("Initiating transaction... (please wait)");

  var meta;
  EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.getMeasurementsOfASmartMeter(id, {from: account});
  }).then(function(value) {
    self.setStatus("Transaction complete!");
    var aux2 = value.valueOf();
    for(var i = 0; i < aux2.length; i++){
      aux2[i]/=decimals;
    }

    var aux1 = document.getElementById("measurementsOfSM");
    aux1.innerHTML = self.makeTableHTML(aux2);
  }).catch(function(e) {
    console.log(e);
    self.setStatus("Error getting measurements; see log.");
  });
},

makeTableHTML: function(myArray) {
  var result = "<table border=1>";
  for(var i=0; i<myArray.length; i++) {
    result += "<tr>";
    result += "<td>"+"hour "+(i+1)+"</td>";
    result += "<td>"+myArray[i] + " kWh"</td>";

  }
  result += "</table>";

  return result;
},

setPriceOfECCoins: function() {

```



```

var self = this;

var price = document.getElementById("priceEC").value * euroToWei;

this.setStatus("Initiating transaction... (please wait)");

var meta;
EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.setPriceOfECCoins(price, {from: account});
}).then(function() {
    self.setStatus("Transaction complete!");
    self.refreshPrices();
}).catch(function(e) {
    console.log(e);
    self.setStatus("Error setting price of EC coins; see log.");
});
},

setContractedPower: function() {
    var self = this;

    var price =
document.getElementById("pricePower").value*decimals/24;
    var powerCont =
document.getElementById("PowerContracted").value*decimals;

    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.setContractedPower(powerCont, price, {from:
account});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshPowerContracted();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error setting price of EC coins; see log.");
    });
},

addNewUser: function() {
    var self = this;

    var user = document.getElementById("userAddress").value;
    var id = document.getElementById("id").value;

    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.addNewUser(user, id, {from: account});
    }).then(function() {

```

```

        self.setStatus("Transaction complete!");
        self.refreshNumberOfUsers();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error adding user; see log.");
    });
},

addNewSmartMeter: function() {
    var self = this;

    var id = document.getElementById("smartMeterid").value;
    var typeSM = document.getElementById("smartMeterType").value;
    var fromuser = document.getElementById("SMfromUser").value;

    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.addNewSmartMeter(id, typeSM, fromuser, {from:
account, gas: 300000});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshNumberOfSmartMeters();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error adding smart meter; see log.");
    });
},

defineDSO: function() {
    var self = this;
    //Next three lines are used to initialized timing for simulation
in the storage of the PC since in the prototype this
//function is only called once and before starting simulating
    var hourNonReset = localStorage.getItem("hourNonReset");
    hourNonReset = 0;
    localStorage.setItem("hourNonReset", hourNonReset);

    var address = document.getElementById("DSOAddress").value;

    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.defineDSO(address, {from: account});
    }).then(function() {
        self.setStatus("Transaction complete!");
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error defining DSO address; see log.");
    });
},

```

```

buyECCoins: function() {
    var self = this;

    var quantity = document.getElementById("numberECToBuy").value;
    var priceEC = document.getElementById("priceOfECCoins").innerText
    * euroToWei;
    var valueTransaction = quantity * priceEC;
    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.buyECCoins(quantity, {from: account, value:
valueTransaction, gas: 300000});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshBalanceEC();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error buying EC coins; see log.");
    });
},

buySCPTokens: function() {
    var self = this;
    var quantity = document.getElementById("numberSCPToBuy").value;
    this.setStatus("Initiating transaction... (please wait)");
    var meta;

    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.buySCPTokens(quantity, {from: account, gas:
300000});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshBalanceSCP();
        self.refreshBalanceEC();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error buying SCP coins; see log.");
    });
},

givePermission: function() {
    var self = this;

    var addressTo = document.getElementById("addressTo").value;
    var quantity = document.getElementById("numberSCPToGive").value;
    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.givePermission(addressTo, quantity, {from: account,
gas: 300000});
    }).then(function() {

```

```

        self.setStatus("Transaction complete!");
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error giving permission; see log.");
    });
},

buySCPTokensFromOtherUser: function() {
    var self = this;

    var addressFrom = document.getElementById("addressFrom").value;
    var quantity =
document.getElementById("numberSCPToBuyFromUser").value;
    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.buySCPTokensFromOtherUser(addressFrom, quantity,
{from: account, gas: 300000});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshBalanceSCP();
        self.refreshBalanceEC();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error giving permission; see log.");
    });
},

burnECCoins: function() {
    var self = this;

    var quantity = document.getElementById("burnEC").value;
    this.setStatus("Initiating transaction... (please wait)");

    var meta;
    EnergyCommunity.deployed().then(function(instance) {
        meta = instance;
        return meta.burnECCoins(quantity, {from: account, gas: 300000});
    }).then(function() {
        self.setStatus("Transaction complete!");
        self.refreshBalanceEC();
    }).catch(function(e) {
        console.log(e);
        self.setStatus("Error giving permission; see log.");
    });
},

burnSCPTokens: function() {
    var self = this;

    var quantity = document.getElementById("burnSCP").value;
    this.setStatus("Initiating transaction... (please wait)");

    var meta;

```

```

EnergyCommunity.deployed().then(function(instance) {
  meta = instance;
  return meta.burnSCPTokens(quantity, {from: account, gas:
300000});
}).then(function() {
  self.setStatus("Transaction complete!");
  self.refreshBalanceSCP();
}).catch(function(e) {
  console.log(e);
  self.setStatus("Error giving permission; see log.");
});
},

withdraw: function() {
  var self = this;

  this.setStatus("Initiating transaction... (please wait)");

  var meta;
  EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.withdraw({from: account, gas: 300000});
  }).then(function() {
    self.setStatus("Transaction complete!");

  }).catch(function(e) {
    console.log(e);
    self.setStatus("Error withdrawing; see log.");
  });
},

events: function(){
  var self = this;
  var ul = document.getElementById("listEvents");
  ul.innerHTML = "";
  var meta;
  EnergyCommunity.deployed().then(function(instance) {
    meta = instance;
    return meta.allEvents({fromBlock: 0, toBlock: 'latest'});
  }).then(function(value){
    var events = value;
    events.watch(function(error, result){
      //console.log(result);
      var node = document.createElement("LI");
      if(result.event == "NewUserCreated"){
        var textnode = document.createTextNode(result.event + ": id=
"+ result.args.UserId + ", address= " + result.args.UserAddress + " ;
block number: " + result.blockNumber);
      }else if(result.event == "AllSCPTokensSold"){
        var textnode = document.createTextNode(result.event + " ;
block number: " + result.blockNumber);
      }else if(result.event == "UserRemoved"){
        var textnode = document.createTextNode(result.event + ": id=
"+ result.args.UserId+ " ; block number: " + result.blockNumber);

```

```

    }else if(result.event == "NewSmartMeterCreated"){
      var textnode = document.createTextNode(result.event +": id=
"+ result.args.SMId + ", type= " + result.args.Type +"; block number:
" + result.blockNumber);
      }else if(result.event == "NewMeasurement"){
        var textnode = document.createTextNode(result.event +":
community="+ result.args.Comm/decimals +"kW , d1= " +
result.args.D1/decimals + "kW , d2= " + result.args.D2/decimals + "kW
, d3= " + result.args.D3/decimals + "kW , PV= " +
result.args.PV/decimals + "kW , Batt= " + result.args.Batt/decimals
+"kW ; block number: " + result.blockNumber);
      }else if(result.event == "NewMarketPrice"){
        var textnode = document.createTextNode(result.event +":
PoolPrice= "+ result.args.PoolPrice/decimals +"€/kWh , AccessTariff= "
+ result.args.AccessTariff/decimals + "€/kWh , NetworkFee= " +
result.args.NetworkFee/decimals +"€/kWh ; block number: " +
result.blockNumber);
      }else if(result.event == "LiquidationDone"){
        var textnode = document.createTextNode(result.event +":
Value= " + result.args.Value/decimals +"EC, From: "+ result.args.From
+" , To: " + result.args.to + "; block number: " +
result.blockNumber);
      }else if(result.event == "PowerContractedPayment"){
        var textnode = document.createTextNode(result.event +":
Value= " + result.args.PowerPayment/decimals + " EC, User 1
contribution: " + result.args.Percentage1/10 +" %, User 2
contribution: "+ result.args.Percentage2/10 +" %, User 3 contribution:
"+ result.args.Percentage3/10 +"% ; block number: " +
result.blockNumber);
      }
    }

    node.appendChild(textnode);
    var list = document.getElementById("listEvents");
    list.insertBefore(node, list.firstChild);

  });
});
}

};

window.addEventListener('load', function() {
  // Checking if Web3 has been injected by the browser (Mist/MetaMask)
  if (typeof web3 !== 'undefined') {
    console.warn("Using web3 detected from external source. If you
find that your accounts don't appear or you have 0 MetaCoin, ensure
you've configured that source properly. If using MetaMask, see the
following link. Feel free to delete this warning. :)
http://truffleframework.com/tutorials/truffle-and-metamask")
    // Use Mist/MetaMask's provider
    window.web3 = new Web3(web3.currentProvider);
  } else {
    console.warn("No web3 detected. Falling back to
http://127.0.0.1:9545. You should remove this fallback when you deploy

```

```
live, as it's inherently insecure. Consider switching to Metamask for
development. More info here:
http://truffleframework.com/tutorials/truffle-and-metamask");
// fallback - use your fallback strategy (local node / hosted node
+ in-dapp id mgmt / fail)
    window.web3 = new Web3(new
Web3.providers.HttpProvider("http://127.0.0.1:9545"));
}

App.start();
});
```

### 10.1.9 INDEX.HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Energy Community Dapp</title>
  <link
href='https://fonts.googleapis.com/css?family=Open+Sans:400,700'
rel='stylesheet' type='text/css'>
  <script src='./app.js'></script>
</head>
<body>
  <div id="header">
    <h1>Energy Community Dapp</h1>
    <br><span id="status"></span>
    <br>
    <br>
  </div>
  <div id="subheaderIzq">
    <p> Total # of SCP tokens: &nbsp;<span class="black"><span
id="numberOfSCPTokens"></span></span></p>
    <p> # of SCP tokens remaining: <span class="black"><span
id="numberOfSCPTokensRemaining"></span></span></p>
  </div>
  <div id="subheaderCenter">
    <p> 1 EC coins = <span class="black"><span
id="priceOfECCoins"></span> €</span></p>
    <p> 1 SCP token = <span class="black"><span
id="priceOfSCPTokens"></span> EC</span></p>
  </div>
  <div id="subheaderRight">
    <button id="start" onclick="App.startSimulation()">START</button>
    <button id="stop" onclick="App.stopSimulation()">STOP</button>
    <button id="reset" onclick="App.resetSimulation()">RESET</button>
    <br>
    <p> Contracted Power = <span class="black"><span
id="ContPowerComm"></span> kW</span></p>
  </div>
```

```

<!-- Tab links -->
<div class="tab">
  <button class="tablinks" onclick="App.openTab(event,
'Owner')">Owner</button>
  <button class="tablinks" onclick="App.openTab(event,
'User')">User</button>
  <button class="tablinks" onclick="App.openTab(event,
'Events')">Events</button>
  <button class="tablinks" onclick="App.openTab(event,
'Graphs')">Graphs</button>
</div>

<!-- Tab content -->
<div id="Owner" class="tabcontent">

  <div id="body">

    <div id="left">
      <h3>Mint SCP tokens</h3>
      <input type="text" id="amount" placeholder="amount: "></input>
      <input type="text" id="priceSCP" placeholder="price in EC:
"></input>
      <button id="send1" onclick="App.mintSCPTokens()">Mint SCP
tokens</button>

      <h3>Set price of EC coins</h3>
      <input type="text" id="priceEC" placeholder="New price in €:
"></input>
      <button id="send2" onclick="App.setPriceOfECCoins()">Set price
of EC coins</button>

      <h3>Add new user</h3>
      <input type="text" id="userAddress" placeholder="User address:
"></input>
      <input type="text" id="id" placeholder="id: "></input>
      <button id="send" onclick="App.addNewUser()">Add user</button>

      <h3>Add new smart meter</h3>
      <input type="text" id="smartMeterid" placeholder="id:
"></input>
      <input type="text" id="smartMeterType" placeholder="bool:
"></input>
      <input type="text" id="SMfromUser" placeholder="from user:
"></input>
      <button id="send" onclick="App.addNewSmartMeter()">Add
SM</button>

      <h3>define DSO address</h3>
      <input type="text" id="DSOAddress" placeholder="DSO address:
"></input>
      <button id="send" onclick="App.defineDSO()">Apply</button>

```



```

<h3>Withdraw from Smart Contract</h3>
<button id="send3" onclick="App.withdraw()">Withdraw</button>

<h3>Burn EC coins</h3>
<input type="text" id="burnEC" placeholder="Quantity to burn:
"></input>
<button id="send4" onclick="App.burnECCoins()">Burn
EC</button>

<h3>Burn SCP tokens</h3>
<input type="text" id="burnSCP" placeholder="Quantity to burn:
"></input>
<button id="send4" onclick="App.burnSCPTokens()">Burn
SCP</button>
</div>

<div id="rightTop">
  <h3>You have <span class="black"><span
id="balanceECOwner"></span> EC coins</span></h3>
  <h3>You have <span class="black"><span id =
"balanceSCPOwner"></span> SCP tokens</span></h3>
</div>

<div id="rightBottom">
  <h3>Power contracted options:</h3>
  <p> Price contracted Power = <span class="black"><span
id="ContPowerPrice"></span> €/kW day</span></p>
  <input type="text" id="PowerContracted" placeholder="Cont.
Power (kW): "></input>
  <input type="text" id="pricePower" placeholder="Price €/kW
day: "></input>
  <button id="send"
onclick="App.setContractedPower()">Apply</button>
  <br><br>
  <p> max power = <span class="black"><span
id="maxPowerInOrOut"></span> kW </span></p>

</div>
</div>

</div>

<div id="User" class="tabcontent">
  <div id="body">

    <div id="left">
      <h2>USER <span id="user"></span></h2>
      <br><select id="userList" onchange="App.userChange()">
        <option selected value="0"> Choose one option </option>
        <option value="1"> User 1 </option>
        <option value="2"> User 2 </option>
        <option value="3"> User 3 </option>
      </select>
    </div>
  </div>
</div>

```

```

</select>

<p> Number of users = <span class="black"><span
id="usersLength"></span></span></p>
<p> Number of Smart Meters = <span class="black"><span
id="numberOfSM"></span></span></p>

<h3>Buy EC coins</h3>
<input type="text" id="numberECToBuy" placeholder="# EC coins to
buy: "></input>
<button id="send2" onclick="App.buyECCoins () ">Buy</button>
<h3>Buy SCP tokens</h3>
<input type="text" id="numberSCPToBuy" placeholder="# SCP tokens
to buy: "></input>
<button id="send2" onclick="App.buySCPTokens () ">Buy</button>
<h3>Give permission to buy SCP to other user: </h3>
<input type="text" id="addressTo" placeholder="give permission
to address: "></input>
<input type="text" id="numberSCPToGive" placeholder="# SCP
tokens to give: "></input>
<button id="send2" onclick="App.givePermission () ">Give
permission</button>
<h3>Buy SCP from other user</h3>
<input type="text" id="addressFrom" placeholder="Buy from
address: "></input>
<input type="text" id="numberSCPToBuyFromUser" placeholder="#
SCP tokens to buy: "></input>
<button id="send2"
onclick="App.buySCPTokensFromOtherUser () ">Buy</button>

</div>

<div id="rightTop">
<h3>You have <span class="black"><span
id="balanceECUser"></span> EC coins</span></h3>
<h3>You have <span class="black"><span id =
"balanceSCPUser"></span> SCP tokens</span></h3>
</div>

<div id="rightBottom">
<p>Balance Instantaneo user 1: <span
id="balanceInstantaneo1"></span> EC</p>
<p>Balance Instantaneo user 2: <span
id="balanceInstantaneo2"></span> EC</p>
<p>Balance Instantaneo user 3: <span
id="balanceInstantaneo3"></span> EC</p>

<p>Balance Instantaneo DSO: <span
id="balanceInstantaneoDSO"></span> EC</p>
<button id="send2"
onclick="App.liquidationOfTransactions () ">liquidate</button>
<br>
<h3>Get measurements of a smart meter</h3>
<input type="text" id="idSMTToShow" placeholder="id of SM:
"></input>

```

```

        <button id="send"
onclick="App.getMeasurementsOfASmartMeter()">get measurements</button>
        <span class="black">Measurements of SM <span
id="idOfSm"></span>: <br><span id="measurementsOfSM"></span></span>
        <br>

    </div>
</div>

</div>

<div id="Events" class="tabcontent">
    <h3>Events</h3>
    <p><span id="events"></span> </p>
    <ul type="1" id="listEvents">

    </ul>

</div>

<div id="Graphs" class="tabcontent">
    <div id="body">
        <h3>Plots</h3>
        <input type="text" id="idUSERToShow" placeholder="id of user:
"></input>
        <input type="text" id="fromh" placeholder="from hour: "></input>
        <input type="text" id="toh" placeholder="to hour: "></input>
        <button id="send" onclick="App.plot()">plot</button>
        <br> <br>
        <span class="black">Current hour: <span
id="currentHour"></span></span>

        <div class="chart-container" style="position: relative;
height:80vh; width:80vw">
            <canvas id="demand"></canvas>
        </div>

        <div class="chart-container" style="position: relative;
height:80vh; width:80vw">
            <canvas id="commPVBat"></canvas>
        </div>
        <div class="chart-container" style="position: relative;
height:80vh; width:80vw">>
            <canvas id="economictransactions"></canvas>
        </div>
        <div class="chart-container" style="position: relative;
height:80vh; width:80vw">
            <canvas id="instBal"></canvas>
        </div>

    </div>
</div>

```

```
</body>  
</html>
```

### 10.1.10 APP.CSS

---

```
body {  
  margin-left: 10%;  
  margin-right: 10%;  
  margin-top: 5%;  
  font-family: "Open Sans", sans-serif;  
}  
  
div#header{  
  height: 50px;  
}  
  
div#subheaderIzq{  
  float: left;  
  height: 100px;  
  width: 33%;  
}  
  
div#subheaderCenter{  
  float: left;  
  height: 100px;  
  width: 33%;  
}  
  
div#subheaderRight{  
  height: 100px;  
}  
  
div#body{  
  height: 500px;  
}  
  
div#left{  
  float: left;  
  width: 50%;  
  height: 100%;  
}  
  
div#leftTop{  
  float: left;  
  width: 50%;  
  height: 300px;  
}  
  
div#leftBottom{  
  float: left;  
  width: 50%;
```

```
    height: 50%;
}

div#rightTop{
    float: right;
    width: 50%;
    height: 300px;
}

div#rightBottom{
    float: right;
    width: 50%;
    height: 50%;
}

div#text{
    width: 250px;
    background-color: red;
}

div#instance{
    float:right;
    background-color: green;
}

tabl{
    padding-left: 10px;
}

label {
    display: inline-block;
    width:300;
}

p{
    width: auto;
    color: #AAA;
}

input {
    width: 110px;
    padding: 5px;
    font-size: 16px;
    margin-bottom: 2px;
}

button {
    font-size: 16px;
    padding: 5px;
    cursor: pointer;
}

h1, h2 {
    display: inline-block;
    vertical-align: middle;
```

```
margin-top: 0px;
margin-bottom: 10px;
}

h2 {
color: #AAA;
font-size: 32px;
}

h3 {
font-weight: normal;
color: #AAA;
font-size: 24px;
}

.long{
width: 500px;
}

.short{
width: 100px;
}

.black {
color: black;
}

#balance {
color: black;
}

.hint {
color: #666;
}

/* Style the tab */
.tab {
overflow: hidden;
border: 1px solid #ccc;
background-color: #f1f1f1;
}

/* Style the buttons that are used to open the tab content */
.tab button {
background-color: inherit;
float: left;
border: none;
outline: none;
cursor: pointer;
padding: 14px 16px;
transition: 0.3s;
}

/* Change background color of buttons on hover */
.tab button:hover {
background-color: #ddd;
}
```

```

}

/* Create an active/current tablink class */
.tab button.active {
    background-color: #ccc;
}

/* Style the tab content */
.tabcontent {
    display: none;
    padding: 3px 6px;
    border: 1px solid #ccc;
    border-top: none;
    height: 1000px;
}

```

## 10.2 ANNEX B. LAYOUT OF THE WEB APP

Screenshot 1. The owner is about to sign a transaction that will emit 770 SCP tokens with a price of 100 EC coins

The screenshot shows a web application interface with a navigation menu (Owner, User, Events, Graphs) and several sections:

- Mint SCP tokens:** Includes input fields for '770' and '100', and a 'Mint SCP tokens' button.
- Set price of EC coins:** Includes a 'New price in €:' label, an input field, and a 'Set price of EC coins' button.
- Add new user:** Includes 'User address:' and 'id:' labels, input fields, and an 'Add user' button.
- Add new smart meter:** Includes 'id:' and 'bool:' labels, input fields, a 'from user:' label, and an 'Add SM' button.
- define DSO address:** A text input field.

On the right side, there is a summary of the user's assets:

- You have 250000 EC coins
- You have 0 SCP tokens

Overlaid on the interface is a MetaMask transaction confirmation window titled 'MetaMask Notification' on the 'Private Network'. It displays the following details:

- CONFIRM TRANSACTION**
- Account 1: 6eD646...e790 (99.446 ETH, 53003.91 USD) > 3CC375...0371
- Amount: 0 ETH, 0.00 USD
- Gas Limit: 155983 UNITS
- Gas Price: 4 GWEI
- Max Transaction Fee: 0.000623 ETH, 0.33 USD
- Max Total: 0.000623 ETH, 0.33 USD
- Data included: 68 bytes
- Buttons: RESET (orange), SUBMIT (green), REJECT (red)

Screenshot 2. After the emission of SCP tokens the app tabs are refreshed. All SCP initially belong to the Owner. The total and remaining SCP tokens of the community are refreshed as well as their price

## Energy Community Dapp

Total # of SCP tokens: 770

1 EC coins = 0.1 €

START STOP RESET

# of SCP tokens remaining: 770

1 SCP token = 100 EC

Contracted Power = 4.6 kW

Owner	User	Events	Graphs
Mint SCP tokens		You have 250000 EC coins	
amount:	price in EC:	You have 770 SCP tokens	
Mint SCP tokens			



Screenshot 3. User 1 is about to buy 60,000 EC coins from the Owner of the community. This will cost him 6 Ethers or 6,000 € according to the Price of EC coins. In this screenshots it can also be seen other features as the current number of users and Smart meter in the community, or other functions the user can use.

The screenshot displays a web application interface with the following elements:

- Navigation Tabs:** Owner, User (selected), Events, Graphs.
- User Profile:** USER 1, with a dropdown menu showing 'User 1'.
- Community Stats:** Number of users = 3, Number of Smart Meters = 6.
- Buy EC coins:** Input field containing '60000' and a 'Buy' button.
- Buy SCP tokens:** Input field containing '# SCP tokens to' and a 'Buy' button.
- Give permission to buy SCP to other user:** Input field containing 'give permission', input field containing '# SCP tokens to', and a 'Give permission' button.
- MetaMask Notification (Inset Window):**
  - Title:** MetaMask Notification
  - Header:** CONFIRM TRANSACTION (Private Network)
  - Transaction:** User 1 (6e1F83...3f74) sending 100,000 ETH (53121.00 USD) to 3CC375...0371.
  - Amount:** 6,000 ETH (3187.26 USD)
  - Gas:** 300000 UNITS
  - Gas Limit:** 300000 UNITS
  - Gas Price:** 4 GWEI
  - Max Transaction Fee:** 0.001200 ETH (0.64 USD)
  - Max Total:** 6.001 ETH (3187.90 USD)
  - Data included:** 36 bytes
  - Buttons:** RESET, SUBMIT, REJECT

*Screenshot 4. Right after, User 1 buys 10% of the total number of SCP tokens, so 77. For it, he pays 7,700 EC coins attending to the price. His balances are refreshed as well as the SCP tokens remaining in the community to fund the installation of the self-consumption technologies.*

Total # of SCP tokens: 770      1 EC coins = 0.1 €     

# of SCP tokens remaining: 693      1 SCP token = 100 EC      Contracted Power = 4.6 kW

Owner	User	Events	Graphs
<b>USER 1</b>			
<input style="width: 100%;" type="text" value="User 1"/>		You have 52300 EC coins	
Number of users = 3		You have 77 SCP tokens	
Number of Smart Meters = 6			

Screenshot 5. After users have bought EC coins for buying SCP tokens and to be able to participate in the platform, the Owner clicks on Withdraw and will collect all the Ethers that have been paid to the Smart contract.

The screenshot shows a web interface with several sections:

- Add new smart meter:** Includes input fields for 'id:', 'bool:', and 'from user:', along with an 'Add SM' button.
- define DSO address:** Includes an input field for 'DSO address:' and an 'Apply' button.
- Withdraw from Smart Contract:** Includes a 'Withdraw' button.
- Burn EC coins:** Includes an input field for 'Quantity to burr' and a 'Burn EC' button.
- Burn SCP tokens:** Includes an input field for 'Quantity to burr' and a 'Burn SCP' button.

On the right side, there is a summary of contract parameters:

Price contracted Power = 0.1159992 €/kW day

Cont. Power (k) | Price €/kW day: | Apply

Overlaid on the interface are two MetaMask windows:

- MetaMask Notification - CONFIRM TRANSACTION:** Shows a transaction on the Private Network. The 'Owner' (6eD646...e790) is sending 99.442 ETH (51709.32 USD) to address 3CC375...0371. Transaction details include: Amount: 0 ETH (0.00 USD), Gas Limit: 300000 UNITS, Gas Price: 4 GWEI, Max Transaction Fee: 0.001200 ETH (0.62 USD), and Max Total: 0.001200 ETH (0.62 USD). Buttons for 'RESET', 'SUBMIT', and 'REJECT' are visible.
- MetaMask Profile - Owner:** Shows the profile for the 'Owner' (6eD646...e790) with a balance of 123.442 ETH (64189.02 USD). Buttons for 'BUY' and 'SEND' are visible.

Screenshot 6. In hour 14, user 2 agrees with user 1 to transfer him 250 SCP tokens. In order to do so, he liberates this amount of tokens to User 1 so that he can complete the purchase. He enters User 1 address and the amount he is liberating and signs the transaction

**Buy SCP tokens**

# SCP tokens to

**Give permission to buy SCP to other user:**

**Buy SCP from other user**

MetaMask Notification

**CONFIRM TRANSACTION** Private Network

**User 2**  
76Fe8F...69F2  
93.998 ETH  
49019.52 USD

>

3CC375...0371

Amount: 0 ETH / 0.00 USD

Gas Limit:  UNITS

Gas Price:  GWEI

Max Transaction Fee: 0.001200 ETH / 0.63 USD

**Max Total**: 0.001200 ETH / 0.63 USD

Data included: 68 bytes

Screenshot 7. Right after, User 1 enters the account of user 2 and the amount of tokens he is going to buy from him and signs the transaction. He will acquire the SCP tokens in exchange of EC coins

**Buy SCP tokens**

# SCP tokens to

**Give permission to buy SCP to other user:**

**Buy SCP from other user**

MetaMask Notification

**CONFIRM TRANSACTION** Private Network

**User 1**  
6e1F83...3f74  
93.999 ETH  
49019.82 USD

>

3CC375...0371

Amount: 0 ETH / 0.00 USD

Gas Limit:  UNITS

Gas Price:  GWEI

Max Transaction Fee: 0.001200 ETH / 0.63 USD

**Max Total**: 0.001200 ETH / 0.63 USD

Data included: 68 bytes

Screenshot 8. The balance of both users are refreshed. Now User 1 has 250 SCP tokens more reaching the amount of 327 (42.5 %) while User 1 that started with a 70 % of the tokens, 539, now has 289 that is equal to 37.5 %

### USER 1

User 1 ▾

Number of users = 3

Number of Smart Meters = 6

You have 27300 EC coins

You have 327 SCP tokens

### USER 2

User 2 ▾

Number of users = 3

Number of Smart Meters = 6

You have 31100 EC coins

You have 289 SCP tokens

Screenshot 9. Users can check all events that are being uploaded in the Blockchain by going to the “Events” tab. This screenshot shows the events of new measurements and prices that were uploaded in the last hours.

## Energy Community Dapp

Total # of SCP tokens: 770

1 EC coins = 0.1 €

START STOP RESET

# of SCP tokens remaining: 0

1 SCP token = 100 EC

Contracted Power = 4.6 kW

Owner	User	Events	Graphs
Events			
<ul style="list-style-type: none"> <li>• NewMarketPrice: PoolPrice= 0.052€/kWh , AccessTariff= 0.002879€/kWh , NetworkFee= 0.00674€/kWh ; block number: 60</li> <li>• NewMeasurement: community= 0.646kW , d1= 0.431kW , d2= 0.114kW , d3= 0.101kW , PV= 0kW , Batt= 0kW ; block number: 60</li> <li>• NewMarketPrice: PoolPrice= 0.05907€/kWh , AccessTariff= 0.062012€/kWh , NetworkFee= 0.0109€/kWh ; block number: 59</li> <li>• NewMeasurement: community= 0kW , d1= 0.152kW , d2= 0.81kW , d3= 0.118kW , PV= 0kW , Batt= 1.08kW ; block number: 59</li> <li>• NewMarketPrice: PoolPrice= 0.06069€/kWh , AccessTariff= 0.062012€/kWh , NetworkFee= 0.0109€/kWh ; block number: 58</li> <li>• NewMeasurement: community= 0kW , d1= 0.456kW , d2= 0.127kW , d3= 0.1kW , PV= 0kW , Batt= 0.683kW ; block number: 58</li> <li>• NewMarketPrice: PoolPrice= 0.0641999999€/kWh , AccessTariff= 0.062012€/kWh , NetworkFee= 0.0109€/kWh ; block number: 57</li> </ul>			

*Screenshot 10. Finally users can also go to the “Graphs” tab and check their statistics of demand, generation inside the Community, and economic flows in the hours they choose. In the following figures it is shown the statistics for user 1 for the first 24 hours, after having 10 % of SCP participation in the first 14 hours and 42.5 % in the last 10 hours*

## Energy Community Dapp

Total # of SCP tokens: 770

1 EC coins = 0.1 €

START

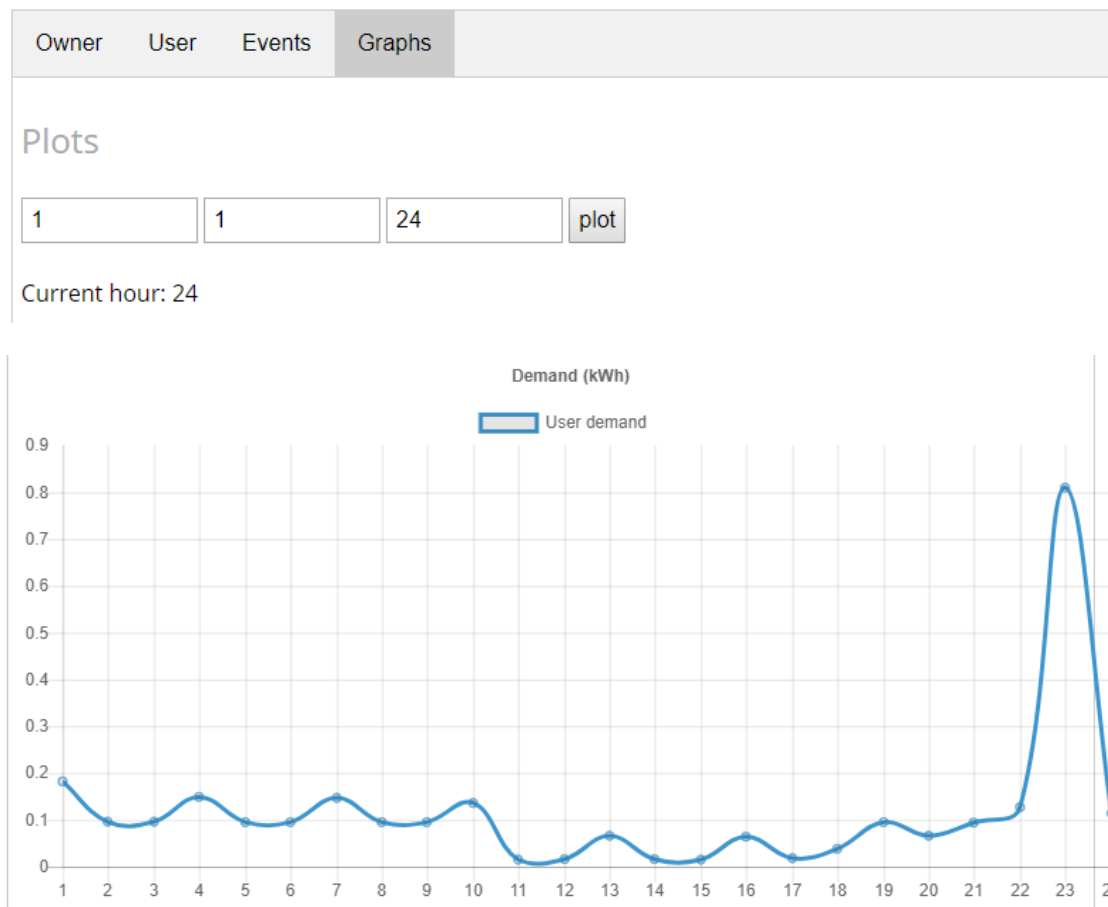
STOP

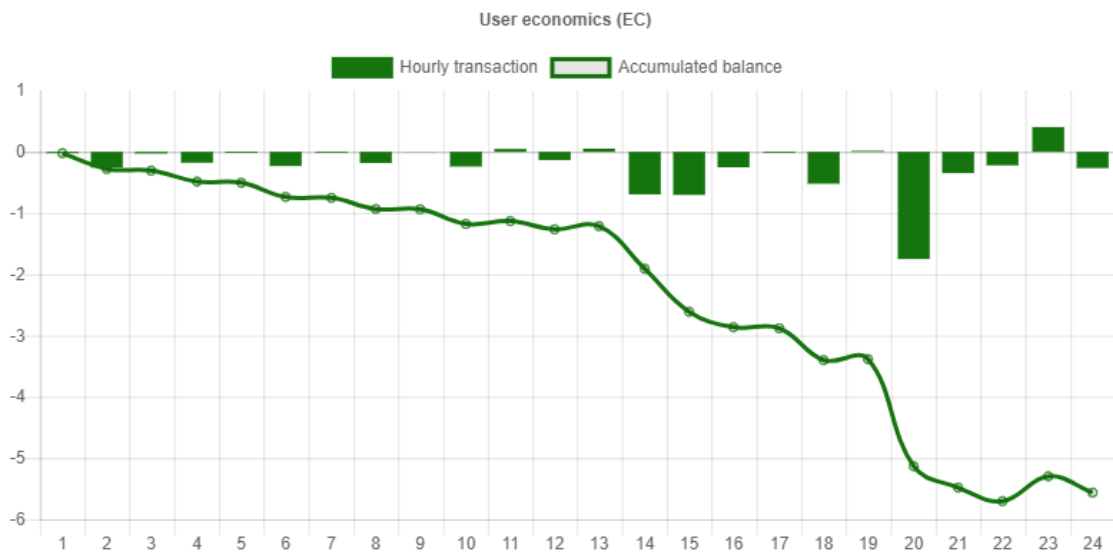
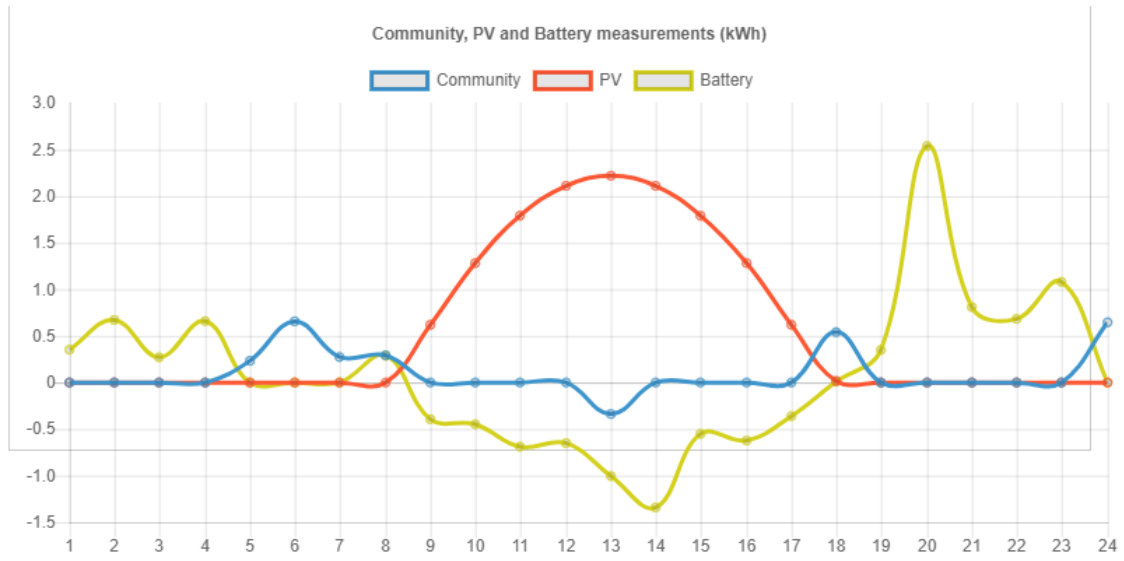
RESET

# of SCP tokens remaining: 0

1 SCP token = 100 EC

Contracted Power = 4.6 kW









## Chapter 11 BIBLIOGRAPHY

- [1] A. Orlov, "Blockchain in the Electricity Market: Identification and analysis of business models," Paris, 2017.
- [2] H. Diedrich, ethereum, Wildfire Publishing, 2016.
- [3] GitHub, "Solidity Documentation," [Online]. Available: <http://solidity.readthedocs.io/en/v0.4.23/>.
- [4] Blockchain.info, "Blockchain.info," [Online]. Available: <https://blockchain.info/es/charts/hash-rate?timespan=all>.
- [5] S. Seth, "Public, Private, Permissioned Blockchains Compared," Investopedia, April 2018. [Online]. Available: <https://www.investopedia.com/news/public-private-permissioned-blockchains-compared/>. [Accessed 2018 May].
- [6] Alastria, "Alastria, consorcio nacional multisectorial blockchain," 2018. [Online]. Available: [https://alastria.io/assets/docs/Alastria\\_Presentacio%CC%81n\\_general\\_.pdf](https://alastria.io/assets/docs/Alastria_Presentacio%CC%81n_general_.pdf).
- [7] J. Monegro, "The Blockchain Application Stack," Coin Desk, 2014. [Online]. Available: <https://www.coindesk.com/blockchain-application-stack/>. [Accessed May 2018].
- [8] A. Preukschat, Blockchain. La revolución industrial de internet, chapter 2, Grupo Planeta, 2017.
- [9] NASA, "NASA, Earth observatory," 2015. [Online]. Available: <https://earthobservatory.nasa.gov/Features/WorldOfChange/decadaltemp.php>. [Accessed May 2018].
- [10] tsp data portal, "The shift project data portal," [Online]. Available: <http://www.tsp-data-portal.org/Energy-Balance-by-Country#tspQvChart>.

- [11] Smart Grids info, "El Parlamento Europeo, a favor de aumentar la cuota de las energías renovables hasta el 35% en 2030," January 2018. [Online]. Available: <https://www.smartgridsinfo.es/2018/01/23/parlamento-europeo-a-favor-aumentar-cuota-energias-renovables-35-2030>. [Accessed May 2018].
- [12] MIT energy initiative, Utility of the future, 2016.
- [13] Irena, "Irena bajada de precios PV," [Online]. Available: [http://www.irena.org/DocumentDownloads/Publications/IRENA\\_Power\\_to\\_Change\\_2016.pdf](http://www.irena.org/DocumentDownloads/Publications/IRENA_Power_to_Change_2016.pdf).
- [14] CE Delft, "The potential of energy citizens in the European Union," 2016.
- [15] CAISO, "What the duck curve tells us about," 2016. [Online]. Available: [http://www.caiso.com/Documents/Flexibleresourceshelprenewables\\_FastFacts.pdf](http://www.caiso.com/Documents/Flexibleresourceshelprenewables_FastFacts.pdf). [Accessed May 2018].
- [16] Bloomberg, "The electric car revolution is accelerating," July 2017.
- [17] CEER, "Renewable Self-Consumers and Energy Communities," July 2017.
- [18] ResCOOP, "Rescoop," [Online]. Available: <https://www.rescoop.eu/>.
- [19] European Commission, "Commission proposes new rules for consumer centred clean energy transition," Nov 2016. [Online]. Available: <https://ec.europa.eu/energy/en/news/commission-proposes-new-rules-consumer-centred-clean-energy-transition>. [Accessed May 2018].
- [20] L. Ojea, "El autoconsumo compartido ya es posible en España: el TC anula la prohibición del Gobierno," *El Periódico de la Energía*, pp. <https://elperiodicodelaenergia.com/el-autoconsumo-compartido-ya-es-posible-en-espana-el-tc-anula-la-prohibicion-del-gobierno/>, Jun 2017.
- [21] R. Roca, "El autoconsumo compartido ya es una realidad en España: Holaluz monta la primera instalación en Barcelona," *El Periódico de la Energía*, Sept 2017.
- [22] Klenergy Tech, "Pylon Network White Paper," [Online]. Available: [https://pylon-network.org/wp-content/uploads/2017/07/170730\\_WP-PYLON\\_EN.pdf](https://pylon-network.org/wp-content/uploads/2017/07/170730_WP-PYLON_EN.pdf).
- [23] M. N. Luke and S. J. L. e. al, "Blockchain in Electricity: A critical review of progress to date," Eurelectric, 2018.

- [24] PWC, "Blockchain, an opportunity for energy producers and consumers?," 2016.
- [25] M. d. C. Romero, "Barreras y oportunidades para el desarrollo de comunidades energéticas sostenibles en España. estudio comparativo con Estados Unidos y Alemania," 2015.
- [26] D. V. e. al, "Blockchain for peer-to-peer energy exchanges: design and recommendations," 2017.
- [27] C. P. e. al, "Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids," 2018.
- [28] B. P. Koirala, "Integrated Community energy Systems," 2017.
- [29] C. Eid, "Towards the design of flexibility management in smart grids: A techno-institutional perspective," 2017.
- [30] Power Ledger, "Power Ledger Whitepaper," [Online]. Available: <https://powerledger.io/media/Power-Ledger-Whitepaper-v3.pdf>.
- [31] "TransActive web page," [Online]. Available: <https://lo3energy.com/>.
- [32] "Conjoule web page," [Online]. Available: <http://www.conjoule.de/de>.
- [33] "Drift web page," [Online]. Available: <https://www.join-drift.com/>.
- [34] "page, Joliette web," [Online]. Available: <https://www.blockchain2business.eu/blockchain-content/2017/12/28/new-blockchain-token-aims-to-decentralize-energy-consumption>.
- [35] "Dajie web page," [Online]. Available: <https://www.rockstart.com/startups/dajie/>.
- [36] "Mybit web page," [Online]. Available: <https://mybit.io/>.
- [37] "Solar Coin web page," [Online]. Available: <https://solarcoin.org/en/node/6>.
- [38] "WePower web page," [Online]. Available: <https://wepower.network/>.
- [39] R. Beck and J. e. a. Czepluch, "BLOCKCHAIN – THE GATEWAY TO TRUST-FREE, Twenty-Fourth European Conference on Information Systems (ECIS),," Copenhagen, 2016.

- [40] M. Opray, "Could a blockchain-based electricity network change the energy market?," *The Guardian*, Jul 2017. [Online]. Available: <https://amp-theguardian-com.cdn.ampproject.org/c/s/amp.theguardian.com/sustainable-business/2017/jul/13/could-a-blockchain-based-electricity-network-change-the-energy-market>.
- [41] Bloomberg, "Bloomberg New Energy Finance 2017," 2017. [Online]. Available: <https://about.bnef.com/new-energy-outlook/>.
- [42] R. Price, "Digital currency Ethereum is cratering because of a \$50 million hack," *Business Insider*, Jun 2016. [Online]. Available: <http://uk.businessinsider.com/dao-hacked-ethereum-crashing-in-value-tens-of-millions-allegedly-stolen-2016-6>.
- [43] A. Hertig, "Ethereum's Big Switch: The New Roadmap to Proof-of-Stake," *Coindesk*, May 2017. [Online]. Available: <https://www.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proof-of-stake/>.
- [44] Pylon Network, "PYLON NETWORK & FAIRCOOP: Construyendo un sistema blockchain escalable para el sector energético," Dic 2017. [Online]. Available: <https://pylon-network.org/es/pylon-network-faircoop-construyendo-un-sistema-blockchain-escalable-para-el-sector-energetico.html>.
- [45] European Commission, "ETRI 2014, Energy Technology Reference Indicator 2010-2050," 2014.
- [46] E. L. L. S. J.M. Fernández de Bobadilla, "Modelo de gestión óptima de baterías: manual de usuario, modelado y validación," 2017.
- [47] JRC European Commission, "PVGIS," [Online]. Available: <http://re.jrc.ec.europa.eu/pvgis/>.
- [48] GitHub, "Solidity Documentation," [Online]. Available: <http://solidity.readthedocs.io/en/v0.4.23/>.
- [49] "Cryptozombies tutorial," [Online]. Available: <https://cryptozombies.io/>.
- [50] "w3Schools," [Online]. Available: <https://www.w3schools.com/>.
- [51] "Truffle," [Online]. Available: <http://truffleframework.com/>.

- [52] "MetaMask," [Online]. Available: <https://metamask.io/>.
- [53] V. P. P. G. e. a. Christopher Jones, "Battery storage for post-incentive PV uptake? A financial and life cycle carbon assessment of a non-domestic building," Elsevier, 2017.

