IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 13, 2021, accepted May 13, 2021, date of publication May 19, 2021, date of current version June 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3081932

Towards Blockchain-Based Internet of Things Systems for Energy Smart Contracts With Constrained Hardware Devices and Cloud Infrastructure

PABLO MÉNDEZ ROYO[1], **JESÚS RODRÍGUEZ-MOLINA**[1], **JUAN GARBAJOSA**[2], **(Senior Member, IEEE), AND PEDRO CASTILLEJO**[1]
[1]Department of Telematics and Electronics Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Higher Technical School of Engineering of Information Systems, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Jesús Rodríguez-Molina (jesus.rodriguezm@upm.es)

ABSTRACT Hardware solutions based on blockchain used in peer-to-peer electricity trading operations have been on rise during the last years. It is expected that due to their usage, it will become easier for prosumers to participate in the power grid on more equal terms when compared to the traditional players that have been settled in this market for the last decades. However, devices used to fully integrate prosumers are scarce and often offer minimal functionalities to perform the task of becoming integrated in those markets. This manuscript puts forward a Constrained Hardware Device enhanced with several software elements related to blockchain and cloud infrastructures, which make possible for any electricity generator or storage system to perform major actions like executing smart contracts, requesting energy prices to a Transmission System Operator and replicating the interchanged data in a cloud computing environment in case there are blockchain node failures. In this way, Renewable Energy Sources can be integrated by means of inexpensive, reliable devices with all the required software components preinstalled, with prosumers being able to further intervene in energy markets.

INDEX TERMS Blockchain, smart contract, energy, Internet of Things.

I. INTRODUCTION

The integration of Internet of Things and smart grid paradigms is leading the transformation of the power supply market across the world. Currently, consumers are able to know their power consumption with a finer than ever resolution due to increasing smart meter precision. For example, consumption per appliance or per hour, instant consumption or daily/hourly consumption can be known in at a very accurate level. Moreover, the consumer role is being transformed to a novel stakeholder: the *prosumer*. The prosumer is a user that not only consumes energy but also produces it by means of a plethora of Renewable Energy Sources (RES): photo-voltaic panels, wind micro turbines, etc. As stated in the 2019 Q4 European Commission report on European Electricity Markets [1] "*in a growing number of countries, renewable sources have reached maturity and can compete with other generation assets on market terms*". In addition, a prosumer can store energy using batteries (or even their own Electric Vehicle or EV) to sell the generated energy when the price is higher. With this new stakeholder, the energy market is expected to pivot from a closed market with only large suppliers selling/buying energy (Energy Service Companies - ESCOs-) to a distributed market with small private prosumers participating in it, which promises many benefits [2].

In order to organize and make this new market completely transparent and auditable, the use of a shared and distributed ledger appears as an ideal solution. Thus, blockchain-based solutions (that essentially make use of a distributed ledger) along with smart contracts are often expected to be the missing piece to complete the new worldwide energy market. The advantages that smart contracts can offer are already widely known; the most evident one is that they are able to describe in an accurate manner the main terms and characteristics of

The associate editor coordinating the review of this manuscript and approving it for publication was Srinivas Sampalli.

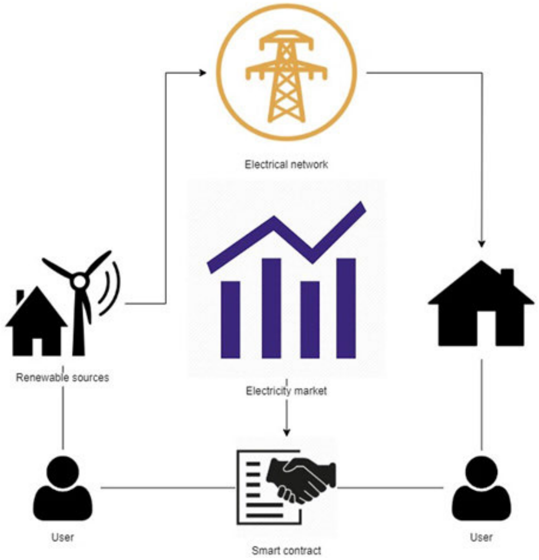


FIGURE 1. Smart contract interactions with all the other parties in energy generation, distribution and consumption.

a good or service that is going to be interchanged, to the point that these features become part of the code that they have been written to. Consequently, much of the ambiguity and delay in contract execution that would take place with regular contracts disappear under this paradigm. While complex data interchange is not anything new in the IoT or in distributed, Cyber-Physical Systems [3], the addition of data that is legally liable is a novelty offered by smart contracts that should not be overlooked by any means. As it can be seen in Figure 1, as far as energy trade is concerned, there are three main actors: end users or prosumers that are able and willing to interact with the energy markets according to business models beneficial to them [4], the electricity markets where trade currently takes place and their Renewable Energy Sources (RES) used to generate the good -in this case energy- that will be traded. These are the participants of the power grid that will be targeted by smart contracts.

This manuscript deals with the potential that blockchain-enabled, hardware constrained devices offer in terms of energy trading on a peer-to-peer basis, as both the Information and Communication Technologies and the power grid that they are attached to have become mature enough in their development to support each other in a realistic manner. By deploying a system with the aforementioned features, it is possible to enable secure, cheaper and accountable remote peer-to-peer trade of electricity among prosumers integrated within the power grid in a reliable and systematic way, so that agreements on the quantity of energy, when or how is going to be transferred, along with the cost that it will represent for the demand side of the trade, can be formalized in an accurate way at each side of the communication. In addition to that, the possibilities that a) cloud facilities offer as a way to have a distributed data backup and b) the nature of smart contracts, that require no central authority to be executed and enforced, have also been taken into account in this paper.

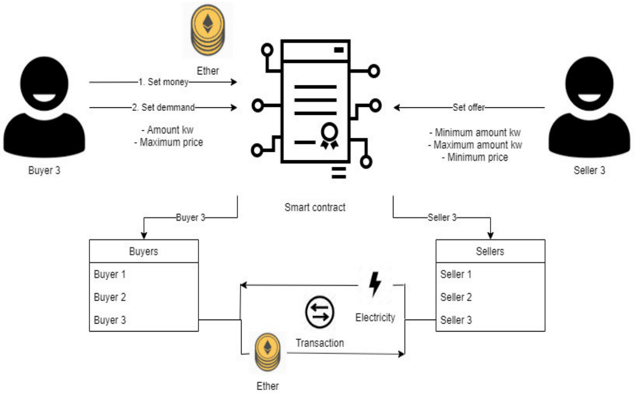


FIGURE 2. Overall structure and performance of smart contracts between sellers and buyers.

A. SYNERGIES BETWEEN THE POWER GRID AND BLOCKCHAIN

Blockchain technology is a distributed data base along a set of nodes. Data is stored in information pieces called transactions. Transactions are grouped in blocks which contain from 1 to N transactions each. Each block of the chain is linked to its previous and next blocks by means of cryptographic mechanisms. The set of blocks is the ledger, where every transaction is stored in an unalterable (immutable) record. This allows every stakeholder to check, review or audit every transaction performed by itself or by any other participant with integrity guarantee. Even regulators can audit the ledger to detect frauds or misuse. Since the ledger is stored in a distributed manner along every node participating, the probability of information loss or spurious attacks is reduced or even impossible.

Smart contracts can be used for the purpose of specifying what a participant is able to register as a transaction (i.e., the operations allowed to participants in a Blockchain). In other words, a smart contract is where the business logic is implemented. Thus, a large utility can use smart contracts to allow their prosumers to only sell a limited amount of energy per day. Another possibility is that a regulator can limit the energy selling/buying range prices by means of smart contracts. Figure 2 shows the basic structure of a smart contract oriented towards the domain of electricity: a) the buyer interested in purchasing electricity sets the amount of electricity that is demanded and the maximum price that they are willing to pay. At the same time, b) the seller establishes the maximum amount of energy that they can sell and the minimum price that they can pay. All these data will be eventually included in a smart contract that, when energy demand meets energy offer, will trigger a transaction between both parties and a trade operation will eventually take place. Many of these kinds of smart contracts run on the Ethereum network since it is easier for start-up companies ([5], [6]) to have their own energy tokens running on top of the system.

The application of blockchain technology to power grids will promote the evolution of smart grids in different aspects. First, energy markets will be changed. As aforementioned, the irruption of new actors (prosumers, private energy cooperatives, energy aggregator SMEs, etc.) has the potential to change completely the energy market across the world. This can be eased by providing a reliable and auditable ledger to harmonize the energy market, which can be achieved using Blockchain and Smart Contracts as the basics of this emerging market. Secondly, the integration of renewable energy storage systems in the power network, such as batteries, will be made possible. If the cost of acquiring and installing a battery to store the excess of generated energy is overpassed by the benefits of selling that energy in the best price timeslot, then it will be attractive for prosumers to engage in energy trading. This will attract other consumers to step in and install their own RES and batteries, converting them in prosumers too. Thirdly, grid decentralization will increase. The decentralization of the power grid has a lot of pros: resilience, adaptability, scalability, etc. There are many key smart grid domains that can be also enhanced using blockchain technology [7]. Finally, the use of a public blockchain where every stakeholder (regulators, large companies, Small and Medium Enterprises, prosumers or citizens) is allowed to audit the energy market will lead to a more transparent and fair market where every transaction (or even every Smart Contract) has the potential of being public and open to consult.

This manuscript takes into account all these features of electricity trade, how it is distributed and how it is paid for by putting forward a lightweight device that we refer to a Constrained Hardware Device (CHD) used to perform all the functionalities related to obtaining energy prices from Transmission System Operator (TSO), deploying a smart contract with several energy features and triggering it whenever there is another party that meets the terms that have been put forward. Besides, it is put forward that it can work in close cooperation with cloud-based infrastructure for redundant data storage in case they are required due to legislation or massive data failures. We have proven that a CHD can be used in trade operations where energy interchanges have to consider real energy prices and have information that must be stored in a distributed way.

B. PAPER CONTRIBUTIONS

In this paper, a proposal for a distributed system that makes use of constrained hardware devices to deploy and execute smart contracts that gather data from a TSO and provide it to a cloud infrastructure has been described. In a more detailed way, the contributions that have been made in this manuscript are listed as follows:

1. **Conceptualization of a Constrained Hardware Devices (CHD) for deployment and execution of smart contracts**. It is shown that smart contracts can be used in combination with hardware that has comparatively low capabilities (in our implementation, Raspberry Pi 3B+ devices, with limited amounts of Random Access Memory, storage and CPU speed) that will also be able to deploy and execute them. This has significant implications: it means that any device capable of storing energy (for example, home or Electric Vehicle batteries) can be plugged or make use of constrained hardware devices to connect to energy markets, obtain transaction prices and execute contracts that will reflect agreements among participants in the network. It also shows that constrained devices can be used as hardware agents with installed software that will grant the terms agreed upon parties to transfer electricity. Finally, all the required hardware and software parts to build a functional Constrained Hardware Device (CHD) are described in this manuscript. They are the cornerstone of the distributed, blockchain-based system that is being put forward.
2. **Usage of CHDs for lightweight consensus algorithm and cloud computing data storage.** Cloud computing facilities can be used to the system's advantage when trying to store information about the smart contracts deployed and executed in the blockchain. It is known that one of the main issues faced by blockchain is scalability: storing all the transactions that have ever taken place in the system can be challenging, due to the sheer amount of information that would have to be eventually saved. By making use of cloud infrastructures, not only a way to double-store information about transactions can be offered, but it can also be ensured that greater amounts of data can be saved without adding strain to the CHDs used for other tasks (contract writing, deployment and execution) in case of hardware breakdown or partial network failure, while keeping these data accessible for the users of the system. To an extent, the main constrain in terms of data will be outsourced from the constrained hardware located in the dependencies of the prosumers to the cloud where data storage capabilities are way better. In addition to that, a) the consensus algorithm used by the device described here is based on Proof-Of-Authority (POA), which is far less demanding in computational resources and is more suitable for constrained hardware and b) the CHD can send information towards cloud computing facilities to create a backup of information regarding how it is saved.
3. **Data retrieval in real time from TSO.** The system that has been built relies on three main features: low capability devices, cloud computing to outsource the data to be stored, and information collection from energy markets. In this system, electricity process can be obtained via Representational State Transfer (REST) calls through the Application Programming Interface (API) that is offered by Red Eléctrica Española ([8]). In this way, it is possible to know at what prices energy was sold in the markets immediately before a smart contract is set, so the system can have accurate figures of prices of energy set by any other market transaction that is not making use of the deployed blockchain, as it is expected to coexist with other systems as well.
4. **Procedure for automated delivery of utility services via smart contracts with no intermediaries.** This

manuscript shows how blockchain-based distributed systems running with the aid of smart contracts can be used for the interchange of utilities among participants in a market. Specifically, Section III has a sequence diagram where the steps taken to enable the energy trade are described. Besides, it must be taken into account that in any system where a public blockchain is used, information about prices of the goods and services traded can be done visible to all the participants with ease, as the price of all the transactions taking place can be seen and analyze, thus removing the need for a central party responsible for providing trust in those figures. In this way, the interchange of goods can be done without any intermediary or broker to receive and resend the trade proposals that the smart contracts represents. The proposed CHD can also be used in other application domains to an extent, as long as a) the interchanged good can be quantified and included in a smart contract and b) it makes sense to use low capability devices.

Finally, it must be considered that rather than just performing a theoretical exercise, the authors of this manuscript have included an implementation of the blockchain-based system that has been put forward here so they can provide the feasibility and performance of their conception and design tasks.

C. PAPER STRUCTURE

This paper is organized as follows. An introduction to the main topics of the manuscript is offered in section I. Section II focuses on the existing state of the art with regards to peer-to-peer transactions that make use of smart contracts for energy interchanges, along with the open issues that have been found in the literature that prompted the creation of the proposed system. Section III describes the system itself with all its components. Section IV shows the demonstrator that has been built to test the hypothetical system that had been conceived before and prove its feasibility in real-like conditions. Section V includes the conclusions and future works that are regarded as the next steps that could be taken in this line of research. A section with the bibliographical references closes this manuscript.

II. RELATED WORKS

The solutions that have been included in this state of the art are the most prominent ones related to the research activities that have been included by the authors of this manuscript. To a greater or lesser extent, they are strongly related to interchange of energy under peer-to-peer systems. However, they seem to show several open issues that have yet to be fully solved.

A. REVIEWED PROPOSALS

A. Laszka *et al.* describe in their own research how forward trading can be used to exchange energy in transactive microgrids [9]. The authors have developed a rigorous solution for this kind of microgrids that addresses privacy, trust, and resilience within a framework of a distributed application platform. Their proposal combines smart contracts and a publish-subscribe middleware, together with a Mixed-Integer Linear Programming (MILP) solver. The addition of middleware, which is frequent in smart grid environments [10] proves to be an advantage for this kind of development, as it makes possible creating patterns for communications, service encasing and Protocol Data Units interchanged at the data level. However, the main focus of the paper is privacy, resilience and trust. While the addition of blockchain greatly contributes to making these goals attainable, little is said about the possibility of using a distributed system with low capability devices making most of the required operations.

Other perspective is introduced by Siano *et al.* [11]. The authors of the manuscript introduce a transactive management infrastructure, proposing the concept of Proof of Energy as a consensus protocol for peer-to-peer (P2P) energy exchanges managed by distributed ledger technology. This infrastructure is claimed to offer the possibility to provide speedy two-way communications. An aggregator is supposed to interface the prosumers that it has been assigned to and with external parties such as the Distributed System Operator (DSO), the Transmission System Operator (TSO) or even the market operators. However, the main purpose of the authors of the manuscript is providing an extensive survey on all the potential solutions that can be offered, rather than providing a more specific solution based on constrained devices that operate within an heterogeneous system.

K. Shuaib *et al.* offer their own perspective on using blockchain for secured and distributed energy exchange [12]. The authors introduced a system supporting energy exchanges between prosumers by means of a kind of smart electronic contracts based on blockchain, implementing a decentralized peer-to-peer energy exchange system. Their work aims to create a decentralized market for buying and selling energy that neither requires the central authority of a qualified entity, nor trust in the other participants of the system. Consequently, it is only natural that blockchain comes in handy as one of the best fitting solutions for this purpose. While the main contributions of our manuscript are more on deploying and executing smart contracts from constrained, low capability devices communicating with other parts of the system, the idea of using a blockchain as a guarantee to provide market decentralization has also been included by us as a natural synergy between markets and blockchain.

Chakraborty *et al.* describe in [13] how the authors of this piece of research have developed an automated peer-to-peer negotiation strategy for settling energy contracts among prosumers in a Residential Energy Cooperative, and how they have considered heterogeneous prosumer preferences. The prosumers engage in bilateral negotiations with peers, seeking for mutual periodical energy contracts/loans. However, this solution is purely aimed to provide a way to negotiate prices for electricity trade in the most optimal possible way. Consequently, the work presented by the authors is lacking a blockchain infrastructure to deploy and save the information related to the trades that take place in it.

Nurgaliev and others put forward in [14] their own blockchain-based decentralized system for energy regulations and exchange, with the objective of avoiding the issues that central authorities may be prone to, such as dependency, while at the same time addressing efficiency, privacy and security. The work done by the researchers proves that blockchain can be introduced in smart grid environments where energy transactions are taking place; the main point of the paper, though, is out of the scope of our manuscript, where a decentralized system with several software and hardware components are described.

Lasla *et al.* [15] presented a blockchain-based energy trading architecture for electric vehicles (EVs) within smart cities, using a smart-contract based trading platform, on top of a private Ethereum network. The system that is put forward works in a similar way that the one that we are presenting here with the exception that the authors are aiming their solution towards making use of Electric Vehicles as the main entities providing power, whereas the system that is presented in our paper targets any kind of piece of equipment that can have a constrained hardware device as an interface between the piece of equipment itself and the decentralized trading infrastructure.

In addition to that, Wang *et al.* [16] have studied distributed energy systems incorporating residential, commercial and industrial users. Energy demand side is supported by a peer-to-peer exchange of information (and energy) in the real-time market, using blockchain to guarantee smart contracts. Authors claim to have obtained economic savings, reduction of peak load and increased market efficiency. Their research work, however, is mostly focused on how power and loads behave through the power grid and/or a smart grid, along with the components that such a system has rather than offering a novel solution based on novel constrained hardware solutions.

In a similar manner, Lin *et al.* [17] introduced a comparative analysis of auction mechanisms and bidding strategies for exchanges in terms of market demand and supply metrics. This analysis is focused on P2P solar electricity. Auction mechanisms considered involve Discriminatory and Uniform k-Double Auction (k-DA). These latter mechanisms are consistent with the idea of building a market for energy where prosumers can participate, but they are not enhanced with the capabilities that blockchain can offer, nor they provide any strategy to use low capability devices to contain the bulk of the software components that have been built for this purpose.

A different kind of work was put forward by A. Amato *et al.* in [18]. In an early work, the authors addressed the multi-agent architecture and interoperability issues, on a Service Level Agreement- (SLA) based negotiation protocol to support a distributed optimization strategy. This procedure could be used for the negotiation of prices in energy exchanges between peers of a neighborhood. However, it was conceived at a time when blockchain was not yet present, so it has been superseded by later developments focused on not using a central-like entity as the trust provider.

Xiaoqi Tan and Yuan Wu offer their own work in [19], which is based on calculating a *Stochastic Shortest Path* (SSP) to find the optimal economic value of lifetime constrained Battery Energy Storage (BES) systems. The authors of the manuscript describe how by using a BES model and an algorithm based on *Layer and Group*, dynamic programming on electricity prices for buying and selling operations can be carried out. The work that the researchers offer is noteworthy in terms of how energy storage plays a role in price formation, but it makes use of solutions out of the scope of this manuscript does not offer any information about how to integrate blockchain with batteries or any other energy storage device via constrained hardware to perform energy trades, either with or without the usage of an intermediary. In addition to that, their model is specifically targeted for situations where price information remains unknown, but in a system running on a blockchain accessible by everyone, prices and transaction features are expected to be known by all the participants in the system. Constrained hardware devices are not considered in this manuscript either.

Long *et al.* [20] put forward as a solution the usage of their system of Aggregated Battery Control for peer-to-peer energy interchanges. In this case, the authors show how it is possible to make use of an aggregated control to optimize the available energy within a community of peers. In the end, they prove that using photovoltaic energy in a community capable of sharing energy among the prosumers that belong to it, electricity sharing will result in a significant reduction in the cost of the energy required by that community. The focus of their work, though, is oriented towards how to share the surplus energy existing from batteries in a community of prosumers, rather than describing what kind of software solution can be used to store transactions in hardware nodes or cloud infrastructure.

Finally, H. Taylor and L. W. Hruska proposed in an earlier work [21] how smart batteries for consumers could be standardized. The authors stress the importance of having mutually understandable data standardized in a certain way so it can be intelligible when information is sent to an entity (such as batteries) to another one. Battery state of charge and some other pieces of information are put forward as data that is required to be known. This piece of work has its importance as a first approach to home batteries and having paved the way for later developments, but it predates in a significant degree the usefulness that can be provided by blockchain and it has not been considered in the manuscript. Other similar concepts (smart contracts) were not taken into consideration either.

B. OPEN ISSUES

After reviewing the solutions found in the literature that resemble the most the kind of developments carried out as depicted here, there is a number of open issues that has been found in the studied proposals. These are as follows:

1. **Lack of focus on constrained hardware.** Many of the solutions that are present in the literature have done great research in terms of price formation, trade or loads consuming energy provided by prosumers, but they do not consider the possibilities that low capability devices can offer at this point. Considering that many of them have grown in resources of RAM and ROM memory or have Central Processing Units (CPUs) of better performance than the ones that were offer just a few years ago, research on how they can be used to integrate components oriented towards blockchain and energy trade (instead of just being used as another kind of Advanced Metering Infrastructure) is a desirable activity.
2. **Decentralization of energy markets is still in its infancy.** Despite the research works that have been carried out in this topic, electricity markets have remained the same for decades: they are still closed to small-sized participants and depend on centralized entities for trade that demands to be trusted in order to participate in the system. In this regard, though, there is an intense activity related to the integration of blockchain into price formation in the market of energy, especially for the one of renewable origin.
3. **Data scalability is too often regarded as a long-term problem.** Although blockchain can offer several advantages in the distributed systems where it is included (data redundancy, reliability, transparency, etc.) its usage often involves other issues; scalability is probably one of the most typical, due to the fact that data tends to be stored and never erased or removed, thus stacking up as time goes by. In the reviewed literature there are very few examples on how to handle large amounts of data in the long term for electricity trade, where to store it and how to access it, which might be quite challenging after several years of using such systems.
4. **Market infrastructure is regarded as a theoretical, homogeneous construct.** When markets are taken into consideration, it usually happens that they are considered to have the same kind of underlying infrastructure (either decentralized via blockchain or centralized as most of them are right now). However, the most likely event to take place is that both kinds of markets, centralized and decentralized, will coexist together during a prolonged amount of time. Very few references in literature reflect this fact.
5. **Presence of smart contracts is irregular**. Although they have been researched on for quite a time, smart contracts are still struggling to be introduced into the energy markets. They are not considered in many energy trade research activities, so their usage in the system is far from being taken for granted.

III. PROPOSED SYSTEM

We have conceived a blockchain-based system that will overcome to a significant extent the open issues that have been found in the literature. As it was mentioned before, the main open issues that have been found are subpar decentralization, lack of focus on constrained hardware, short-sightedness about scalability, non-coexistence of market infrastructure considered and suboptimal integration of smart contracts in energy markets. Therefore, the system to be designed used for energy transactions should have a collection of design requirements that are used as a first step to solve the existing open issues.

It is our purpose in this manuscript to describe a piece of hardware that, when a) combining several hardware and software elements and b) cloud infrastructure as a backup data repository, becomes a novelty by itself. As authors of the manuscript, we believe that this solution tackles to a significant extent the open issues that have been found in the following manner:

1. **Subpar decentralization**: from the data transfer point of view, the device presented here works as a blockchain node, so it does not require -nor acts as- a centralized party providing trust with regards to energy prices. Blockchain makes possible that the information about past trades in terms of energy quantity and price are visible for every participant.
2. **Constrained hardware**: a Raspberry Pi has been used to have the software components installed. This choice was motivated by our experience, as it is the device that we are most familiar with. However, it is used as a container for all the software applications and development that have been developed and installed; many other devices with equivalent computational power could be used instead.
3. **Scalability:** scalability has been improved by taking two already explained actions: choosing Proof-Of-Authority as the consensus algorithm and deploying trade information in cloud computing facilities.
4. **Market trade diversity**: the device presented here is expected to be attached to a source of electricity that will be used for trading operations (buying, selling or storing energy). It can be either a battery or any sort of Renewable Energy Source usable by a small-sized prosumer.

It must be born in mind that having a separate, constrained device capable of performing a series of activities of major importance for energy trade (connection to Transmission System Operator for features as electricity prices, deployment of smart contracts onto a peer-to-peer network, storage of information in a redundant blockchain) that connects to an energy storage facility provides a remarkable degree of portability, greater than whether those features would have to be obtained from the energy storage facility itself. Indeed, the device and the functionalities that run on it (which are implemented exclusively via software) can be reused by any other hardware component for energy storage and trade, provided that the interfaces that are offered by the equipment are implemented.

Therefore, just by interfacing a lightweight device from a Renewable Energy System (RES) those functionalities can be outside the RES itself. While the interfacing must be developed, it becomes much simpler than integrating those functionalities in the RES or its storage capabilities, since developing the interface is easier and demands less work that building those aforementioned functionalities from scratch.

Besides, it must also be considered that although other solutions based on secure network of remote nodes can provide similar functionalities as the ones described here, the authors of this manuscript put forward the usage of a Constrained Hardware Device (CHD) as a fully decentralized node that belongs to a blockchain network that can access cloud computing facilities. This makes CHDs more secure than remote nodes from the design point of view, as no mechanism needs to be added to the node itself to counter the recentralization that results from using a remote point of access to a network.

A. DESIGN REQUIREMENTS

The system to be built should follow the following design requirements:

1. **It should make use of blockchain**. The existence of public energy prices in a distributed database that can be shared by all the participants of a distributed system is one of the main features where blockchain can be used to an advantage. By doing so, the need for a broker or any intermediate party currently used to link offer and demand in the marks is no longer present. This is because any party that wants to know what price should be set for their own electricity trade just has to check the price of the last transaction present in the blockchain (which is shared by every node in the system) and be aware of the existence or lack of demand in electricity at those prices. In addition to that, blockchain can be used as the backbone for smart contracts to be deployed when energy is traded among equals, thus removing any ambiguity with regards to the operation that will take place.
2. **Cloud infrastructure could come as an asset**. If might happen that due to scalability issues the participants in this system are unable to contain in each of the nodes all the information from the system with regards to the transactions that take place since the first time when there is a deployment. In addition to that, hardware-constrained devices are likely to be used for the technology deployed since they can be used to regulate energy flow between the power grid and any Distributed Energy Source depending on their positioning. Consequently, the saved data can be uploaded to the cloud, where storage capabilities are far greater and much more scalable and the constrained hardware devices can be kept focused on contract writing, deployment and execution activities.
3. **Real-time information should be retrieved**. It is not expected that every system will switch to blockchain immediately, but there will be a period of coexistence between blockchain-based and traditional energy markets. Under such circumstances access to data from a centralized party, or at least a party that is taking a central role in the system, can be useful as a way to know the price of the last transactions (or any other operation available) that have taken place. To do so, any API that can be used by the centralized party results advantageous. As mentioned in the introduction section, the existence of REST [22] to offer ways to interface energy prices results of extreme usefulness to have a very specific knowledge on what to expect in offer and demand prices.

TABLE 1. Correspondence between open issues, procedures on how to solve them and what components can be designed.

Open issue	Action to solve it	Resulting component
Automation of energy trades is questionable	Smart contracts can be added to trigger the execution of their conditions (or penalties, if required)	Smart contract
Data scalability is questionable; no backup mechanisms are conceived	Making use of auxiliary infrastructure with larger data storage capabilities	Cloud infrastructure
Decentralization of energy markets data comes as a challenge sometimes	Including blockchain	Blockchain deployment
Low capability devices are usually not considered	Including low capability devices	Constrained hardware
Coexistence of different market infrastructure is not considered	Market heterogeneous infrastructure should be considered	Real-time information via TSO API queries

In this way, most of the open issues that had been found could be dealt with significantly. Besides, as a follow-up of the design requirements that have been established, the components that would result in an implementation of those requirements could be set. Consequently, there can be several pieces of equipment (from the hardware point of view) and software subsystems (from the software one) that would be developed as the implementation of a proposed system solving some of the issues found, so that they can materialize the requirements that have been set before. Table 1 shows what kind of software components have been designed to fulfill requirements that were set in the previous step.

From our point of view, the most suitable way to make use of the combined components that will solve the open issues highlighted in the state of the art is integrating the majority of them in a low capability device that can be defined as a Constrained Hardware Device (CHD). This CHD can be used to comprise many of the hardware and software facilities required to create the distributed system described in this manuscript. The overall appearance of all these components working together has been depicted in Figure 3. Note that the Constrained Hardware Device that is described here is

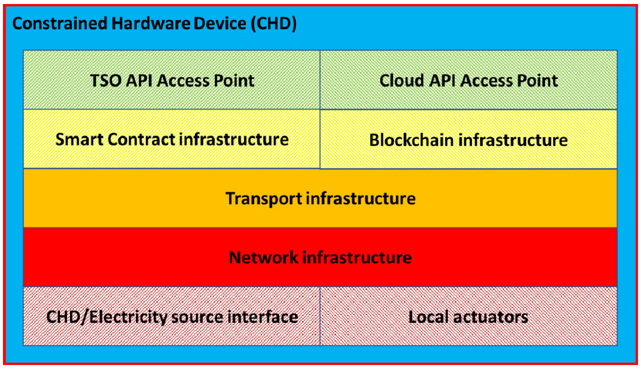


FIGURE 3. Layers within the constrained hardware device.

expected to connect to the electricity infrastructure, which depending on the context could be a Renewable Energy Source, a battery that has been storing energy or, generally speaking, any other appliance capable of producing electricity and injecting it onto a grid, so the CHD as a whole is effectively acting as a gateway between the electricity and the Information and Communication Technology (ICT) domains. The smart contract to be deployed and the blockchain infrastructure have been placed at the same level because they are linked to the same distributed, blockchain-based infrastructure running in each of the nodes where the CHD is used, just like in the higher layer application-based APIs are running.

As it can be seen in the figure, the requirements that have been suggested before have resulted in a collection of specific technologies that fulfill each of them:

1. The lower layer comprises all the hardware elements that are related to both the hardware of the CHD itself and the parts interfacing the electricity domain. That is why there are two different kinds of elements required: on the one hand, the *CHD/Electricity source interface* will take care of regulating the electricity flow between the element attached to the CHD and the power grid itself. Depending on the actions that have been defined in the smart contract, electricity might be transferred or saved until the suitable moment takes place. On the other hand, there must be some parts that show locally whether any kind of energy transaction is taking place. These have been defined as the *Local actuators*, which in a generic way will be responsible for providing information to the end users about the execution of the operations wherever the CHD is located. While their global usefulness in the system is limited, they come in handy as an end user gadget that sends information to the prosumers regarding the execution of the contract that was agreed upon.
2. Further infrastructure will be required to send and receive information through the distributed system that will be created once there are several CHDs installed. In this regard, regular network facilities used according to the TCP/IP (referred to *Transport* and *Network infrastructure* in Figure 3) or OSI architectural models for information transfers will suffice, as the data about smart contracts or other kinds of requests will be encapsulated and treated as packets or segments depending on the level they are transmitted through.
3. At the application layer, but keeping a position different than other software components, the elements related to the decentralized trade of electricity can be found. There are two of them: one is the *Smart Contract infrastructure*, which contains all the aspects related to the smart contract writing (in case a smart contract has to be modified on the fly), deployment (so conditions for electricity trade are known for everyone taking part in the system) and execution (when offer and demand meet at the agreed prices) and the other is the *Blockchain infrastructure* used to interface the data level blockchain network that is participated by the CHD and upon which the smart contracts are deployed and executed. It must be noted that, depending on the deployment, the Blockchain infrastructure encased in the CHD may or may not be eligible to perform further actions related to transaction validation. This procedure resembles the one that is currently used by the Ethereum network, where validation of new transactions and addition of new block is done under a Proof-Of-Stake (POS) algorithm is done by certain nodes of the network that, when providing a stake of cryptocurrencies, become eligible to perform validation, and the greater the stake the higher their chances are to be selected to perform it [23]. As it will be explained in the next section, infrastructure using Proof-Of-Authority as the algorithm was used for development purposes instead.
4. Last but not least, there are two other external entities that must be interfaced in order to have the proposed system working properly in a realistic environment: the Transport System Operator (TSO) and the cloud infrastructure used to save all the data related to the execution of the smart contracts. Consequently, there are two more software components that have been located at the CHD: one is the *TSO API Access point*, used to retrieve information regarding energy prices obtained from the TSO via calls to the API that they usually offer to external entities (as a way to ensure the integration and competitiveness of the proposed blockchain-based system in the electricity markets, which makes use of a different, broker-less technology), and a *Cloud API Access point*, that is pushing information into the cloud for high amounts of data storage. Aside from the nature of their data, the main difference between them is the direction of the information flow: the CHD will tend to "get" data from the TSO (as it contains the information about other trading operations that have taken place in the markets) whereas the overall proposed system will "put" data onto the cloud infrastructure (as it contains the information about the agreed smart contract features).

The sequential procedures followed by the Constrained Hardware Device are explained in the sequence diagram represented in Figure 4. As it can be seen, the CHD A that belongs to the entity with an interest in engaging in electricity trade if

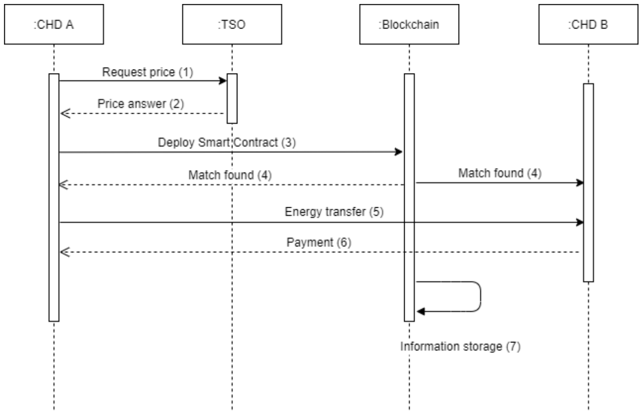


FIGURE 4. **Sequence diagram with the smart contract procedures.**

it is advantageous, sends a request to the Transmission System Operator asking about energy prices (1). Once it receives the answer (TSO, 2) the CHD can use the information to know whether electricity trade is desirable or not (2). If it is so, a smart contract based on the information that has been retrieved will be written down and deployed to the blockchain infrastructure where all the previous data about energy transactions and their costs are stored in a distributed manner (3). Considering the information that has been included in it in terms of energy price and quantity offered, a matching peer entity might be found (CHD B, 4) willing to trade funds for energy in the terms that were published in the smart contract that has been deployed to the network. When that happens, the contract will be triggered, and electricity will be sent from A to B (5) along with the data and payment details associated with the trade (6). Lastly, the information about the transaction that has taken place will be stored in the blockchain where the contract was deployed as well as in the cloud that is being used as a backup for the data interchanges (7). We are aware that, by implementing a solution that makes use of storage in two different repositories (the blockchain itself deployed onto the nodes and the cloud facilities where backup information is kept), additional security measures should be provided to guarantee that data provenance is as expected, and no spurious entity has tampered with it [24]. Such security consideration has been deemed as beyond the scope of what the manuscript is about, as our main interest is presenting what a CHD could do and how the actual implementation of a CHD does it.

As far as the smart contract logic itself is concerned, it has been defined as a regular operation involving selling or buying a good or service and transferring the funds used to do so. Thus, it works in a similar way as the one depicted in Figure 2: a) a contract with a set of characteristics regarding prices (minimum price to sell, maximum price to buy) and a certain amount of energy is deployed onto the blockchain by an interested party (buyer or seller), b) participants in the blockchains enter the system and when there is a match between them c) the smart contract is triggered with the transaction taking place between the two interested parties.

Once the software components have been defined, as well as the kind of hardware that can be used to encase any required program, implementation, deployment and testing works were carried out to have an accurate view of the feasibility of our proposal.

IV. VALIDATION OF THE PROPOSED SYSTEM

In order to prove how the previously described proposal runs actual smart contracts in a distributed environment, a prototype has been built mimicking all the functionalities expected from such a system. Hardware and software components have been used for this purpose, so they will be described in detail for a better grasp of what has been built. The results that have been obtained are oriented towards testing the feasibility of using a CHD as described in this manuscript in order to a) obtain prices in real time from a TSO, b) deploy a smart contract for energy trade among peers, c) make that trade effective and d) store that information onto a blockchain. Next sections describe what components have been used to deploy an actual implementation of a CHD and how information can be transferred.

A. HARDWARE COMPONENTS

If Figure 3 is taken into account, it becomes rapidly apparent that there are three different hardware parts that have to be taken into consideration to build a prototype: the local actuators, the piece of hardware that will be used as the Constrained Hardware Device and the CHD/Electricity source interface that will be used to control the energy flow.

In this regard, the hardware used as CHD is the Raspberry Pi 3B+ [25]. There are several advantages about using this device: a) it has been well-known as a device for prototyping and experimentation for several years already, b) it is sturdy yet inexpensive and easy to install almost everywhere, c) it has all the wired and wireless network interfaces that would be needed to establish connections beyond the local usage of it and d) while its capabilities are constrained (1 GB of RAM memory, flash card for operating system storage, a CPU of 1.4 GHz), they are powerful enough to run the main functionalities expected from a CHD. A board with similar capabilities could have been used as well.

The other pieces of hardware used are extremely simple and easy to obtain. Relays have been used to simulate the flow of electricity into the power grid: when these relays are open, it is considered that electricity goes through the system, and when they are closed its flow is interrupted, thus mirroring what would be done when a smart contract is executed and the CHD as a whole is controlling when energy is transferred. In addition to that, a pair of components made up by a Light Emitting Diode (LED) and a resistor has been used as the actuator providing information to the prosumer about the execution of the smart contract. When the terms (regarding energy offer and demand) of the smart contract that has been deployed in the blockchain are met by another participant in the system, the LED will be turned on to show that a partner to do business with has been found. As far as the proposed system is concerned, no further hardware elements are required for the implementation of the CHD.

B. SOFTWARE ELEMENTS

There is a number of software elements that have been used for the development of a system with the aforementioned features, as integration of several heterogeneous developments is mandatory. There are two kinds of them that can be distinguished: software components demanded from the system and software tools used to implement them.

As far as software components are concerned, they reflect each of the ones that have been shown in Figure 3. They are as follows:

1. **Görli**: this is a testnet (a network mimicking the Ethereum system that can be used for development and prototyping) defined in [26] as "*A cross-client proof-of-authority testing network for Ethereum*". It has been used as the blockchain environment to deploy the smart contracts that are developed and provided by the CHD. As hinted before, this testnet makes use of a Proof-Of-Authority algorithm that, while is close to how POS works (it selects several users deemed to have enough authority to validate transactions the same way some users are chosen in POS), it relies on reputation consensus to solve the energy consumption issues that are usually faced [27].
2. **Infura**: it is used to simulate an Ethereum network node to make communications with the whole blockchain testnet easier [28]. It has been used to abstract the communication between the smart contract deployed and the Görli blockchain network.
3. **Red Eléctrica Española (REE) API**: this component is used in the system to have a way to connect at the data level with the Transmission System Operator (which is REE in the case of Spain) so that information about energy mix composition or prices can be obtained [29]. In this prototype it is used for the CHD to gather a reference price to use so when the smart contract is deployed in the Görli network it is done so at an advantageous cost.
4. **Python scripts**: these are running at the CHD and the cloud infrastructure. They are used for several functionalities: a) one is used for interactions with the energy smart contract, b) another one is used to query the REE API and another one c) contains the configuration settings related to the smart contracts, such as source and destination addresses and d) one more is listening to smart contracts executed. Python has been used as the main programming language, to the point that most of the functionalities required by the CHD are written in Python, except for the deployed smart contract.
5. **Smart contract**: this is one of the main elements of the prototype and the manuscript in general. It sets the conditions for energy selling, such as the price of the electricity quantity that will be sold into the distributed system. It has been written by making use of Solidity, which is one of the most popular languages to write smart contracts.
6. **Amazon Web Services**: they are used to have a reliable, large repository with the data interchanged during energy transactions [30]. In case of the prototype, an *EC2 instance* (EC2 instance is how virtual machines are named in Amazon Web Services) has been created and linked to a MySQL database used for the data storage itself. This EC2 instance has another Python script that listens for new transactions in the energy smart contract and writes them down in the database so data can be kept. In this way, every transaction that takes place in the system will be saved.
7. **Terraform facilities**: the main purpose of these is deploying, managing, and removing cloud infrastructure using scripts of code [31]. They can be regarded as a set of intermediary facilities based on Infrastructure-as-code [32] that eases the access to what has been deployed in the cloud environment, so that the infrastructure saving the data from the transactions can be managed swiftly via programs containing the main features of the previously created EC2 instance.

As for the software tools, they have provided significant usefulness for the development and deployment of the solution:

1. **GitHub**: the most popular software repository up until the moment of writing this manuscript has been used to publish the code and development works that have been used to develop the system described in the previous section [33].
2. **Remix**: this is tool used for writing, debugging, and running smart contracts [34]. It makes use of three plugins (compiler, debugger and a third one for deployment and running transactions) that aided the development of this prototype. It also offers the possibility of deploying a smart contract on a browser with a JavaScript machine and test it while new features are added. Finally, Remix also provides accounts with simulated Ether for testing, which were used for the development of the prototype.
3. **Solidity**: as mentioned before, it is the most popular programming language used for smart contract development [35]. It has been used for exactly that purpose in the prototype.
4. **Truffle and Waffle**: this is a tool that smart contract developers can use to simulate fake environments for testing purposes [36]. In case of this prototype, this software was used for testing the smart contract a local deployment.
5. **Metamask**: this is a browser extension that enables running distributed applications in the Ethereum network without being part of it [37]. To connect to the Ethereum network, Metamask makes use of Infura, a software component shown before. Furthermore, Metamask can be used as a wallet manager, where different accounts can be owned. In this prototype, Metamask was used for managing the accounts and keys used for testing.
6. **Postman**: this is another software program used to test the syntax of requests done against an entity capable of displaying a REST interface that can be queried [38].

As far as the prototype is concerned, it was used to test the interactions between the prototype and the REST API published by REE.

7. **Etherscan**: this is a tool used to check all the information related to Ethereum-based blockchain transactions and participants. As it is claimed in [39], "*Etherscan allows you to explore and search the Ethereum blockchain for transactions, addresses, tokens, prices and other activities*". Data regarding specific transactions and wallets containing different quantities of cryptocurrencies can be seen. Due to the nature of the Ethereum network and blockchain itself, transparency and privacy are the main themes of Etherscan, as it is not possible to link wallet addresses with actual people participating of the network.
8. Other tools, such as **Visual Studio** as the Integrated Development Environment (IDE) were also used to write down the code used for the prototype implementation.

All the software elements used for prototyping development have been included in Table 2 so a summary with their main purposes can be seen.

After all the hardware integration and codification procedures were finished, an actual prototype of the deployed CHDs working in the proposed system was built in the laboratory, as it has been displayed in Figure 5. It must be noted that the deployment is making use of two of the CHDs with all the software features that have been defined in the previous section. Since each of the Raspberry Pi devices used as the CHD is managing a pair of the four relays that have included in the demonstrator, both the Raspberry Pi devices (continuous-lined squares) and their ruling relays (discontinuous-lined squares) have been highlighted in the picture that has been taken.

Once all the components were installed both in the CHD and in the other parts of the blockchain-based distributed system, it could be said that the prototype had been completed. Its overall schematic appearance with all the hardware and software components is shown in Figure 6. Among its remarkable features, the two CHDs were built with all the hardware and software parts so that energy interchanges among them could be carried out by making use of a scenario that mimicked a peer-to-peer energy transaction. This was the chosen use case because it seems as the most likely one to benefit from the distributed system put forward in this manuscript, as CHDs have been conceived to be plugged in other parts of the system. However, there is nothing preventing their usage for other parties that might use larger appliances than just one battery or piece of RES to produce, store or consume energy.

TABLE 2. **Summary of software components and tools used.**

Component	Functionality	Usage
Görli	Offers a blockchain-based testnet with a POA algorithm	Deployment of the written smart contracts
Infura	Simulates Ethereum network node	Communication between the smart contract deployed and Görli
REE API	Offers information about energy prices in specific timeslots by querying it	Collection of electricity prices in a timeslot so smart contracts can be competitive
Python scripts	Versatile programming language	Development of software interactions, smart contract parameters, cloud infrastructure management
Smart contract	Automate buy and sell agreements among parties	Automate buy and sell agreements among parties
Amazon Web Services	Used for varied purposes: data storage, Virtual Machine deployment, higher computational performance	Data storage in a MySQL database, instance for smart contract deployment
Terraform	Infrastructure-as-code. Deploying, managing, and removing cloud infrastructure	Infrastructure-as-code. Deploying, managing, and removing cloud infrastructure
Tool	Functionality	Usage
GitHub	Software repository for project development	Software repository for project development
Remix	Writing, debugging, and running smart contracts	Accounts with simulated Ether. Deployment of smart contracts on a browser with a JavaScript machine
Solidity	Programming language for smart contracts	Programming language for smart contracts
Truffle and Waffle	Simulates fake environments for smart contract execution	Smart contract testing and deployment on a local basis
Metamask	Browser extension that simulates running distributed applications in the Ethereum network. Can be used as a wallet manager	It manages the accounts and keys used for testing.
Postman	Testing of request syntax	Testing of request syntax against the API provided by REE
Etherscan	Freely available information about transactions or accounts	Checked whether transactions were taking place or not
Visual Studio	Integrated Development Environment for programming	Integrated Development Environment for programming

In addition to that, it must be noted that the CHDs are working in close cooperation with the cloud infrastructure that will reflect on every transaction that has taken place within the system, thus using it as a backup every time a transaction takes place. Figure 6 also shows the flow of actions that is taking place in the deployed system: electricity prices for the market transactions that have taken place are requested via REST API to the TSO (1) and a response is obtained with the information (2). With that information, a smart contract is created and deployed to the Görli blockchain network (3) and

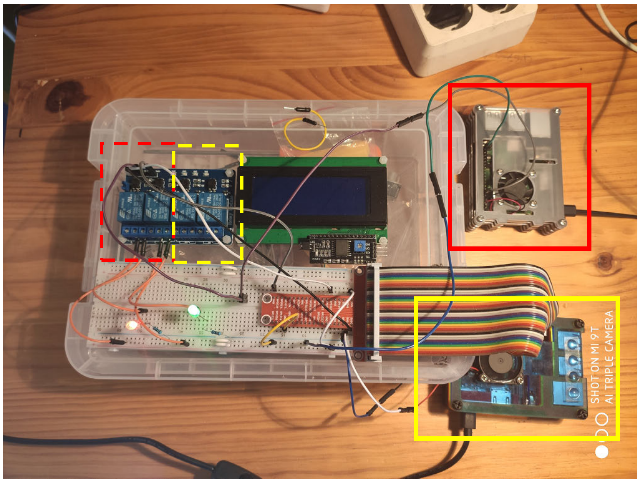


FIGURE 5. **CHDs used as the deployment prototype.**

necessary invocations regarding how buyers and sellers can be set are done by means of a listener thread codified as part of a Python script communicating with Infura (4). At the same time, the cloud infrastructure is running as a Virtual Private Cloud (VPC) listens to any possible transaction through a gateway that is part of the cloud infrastructure itself (5). When a buyer and a seller are matched according to what they offer and what they demand an event transaction will be triggered from the smart contract (6) to both parties and the cloud infrastructure used as a backup. Not only will the battery level vary in both parties because of the smart contract execution (7), but also information about the transactions taking place will be sent to the cloud via gateway (8) and will be written down in the Relational Database (RDS) provided in it (9).

C. SETTING UP THE DEMONSTRATOR AND RESULTS

In this section it has been described how a demonstrator that was built, consisting of two prototypes of CHDs, cloud infrastructure and access to the REST API published by REE, along with all the actions required to deploy and execute a smart contract mimicking electricity trade operation between two peers located in a blockchain-based network. The demonstrator has been used in the following manner to prove that trade operations can take place with the CHD and infrastructure described in this manuscript:

1. Two accounts are created with the Remix tool in order to start with the tests locally. Some simulated Ether is set for both.
2. Since Remix makes possible the deployment of a JavaScript Virtual Machine for smart contract deployments, the machine is set in the laptop that was used for testing purposes.
3. As part of the smart contract to be deployed, there are six functions that are written: *setSeller* (to set the minimum acceptable sell price for the seller and the minimum and maximum amount of KW they would like to sell), *setBuyer* (used to set the maximum amount of energy that someone is willing to buy and its price, provided that the address account of the user has enough funds put in the energy smart contract to buy the amount of electricity specified), *matchSeller* (looks for a seller that matches the requirements of the buyer; called by the *setBuyer* function), *matchBuyer* (looks for a buyer that matches the requirements of the seller; called by the *setSeller* function), *sendFunds* (used to sends funds from a user account to the smart contract; it is like having a wallet inside the energy smart contract, where the user has the funds that will send to the energy smart contract assigned to their account address) and *withdrawFunds* (allows users to retrieve the funds that they have stored in the smart contract).
4. The transaction included in the smart contract is executed in the typical fashion for signed transactions in Ethereum-like networks: since asymmetric cryptography is widely used in blockchain developments [40] the data are signed with the private key of the sender to a receiver, who will use the public key of the sender to verify whether the person who claims to have sent the data are actually them or not. Each smart contract contains several relevant pieces of information, such as a) sender and receiver account addresses, b) digital signature, c) funds, d) start gas and e) gas price. The Blockchain infrastructure of the CHD is used for this purpose, along with the transport and network facilities.
5. After several times, it was deemed that local testing had achieved its objectives, as transactions were working as expected. Therefore, distributed accounts are created by means of the Metamask tool.
6. In order to write the new smart contracts, the price of the last electricity transactions was needed. Thus, the RESTful API published by Red Eléctrica Española (REE) was queried by the TSO API Access Point with the purpose of retrieving the latest available energy trade performed in the markets.
7. The contract is written considering the price obtained from REE. While under a pure blockchain system there would be no need to consult any centralized party about this (the price of the last transaction would be contained as information present in the nodes of the blockchain itself and would be known by everyone) coexistence between blockchain and centralized energy markets has been included.
8. The cloud infrastructure is set. The entity created in AWS is accessed via Terraform infrastructure-as-code capabilities and a script collecting the information about funds transfers is placed for redundancy.
9. The smart contract is deployed and executed in the same way that it had been done locally.

When all is said and done, the trade operations that have been carried out have been completed. To do so, two Constrained Hardware Devices communicating with a Transmission System Operator and cloud infrastructure have deployed a smart contract in a blockchain-like distributed system. Steps

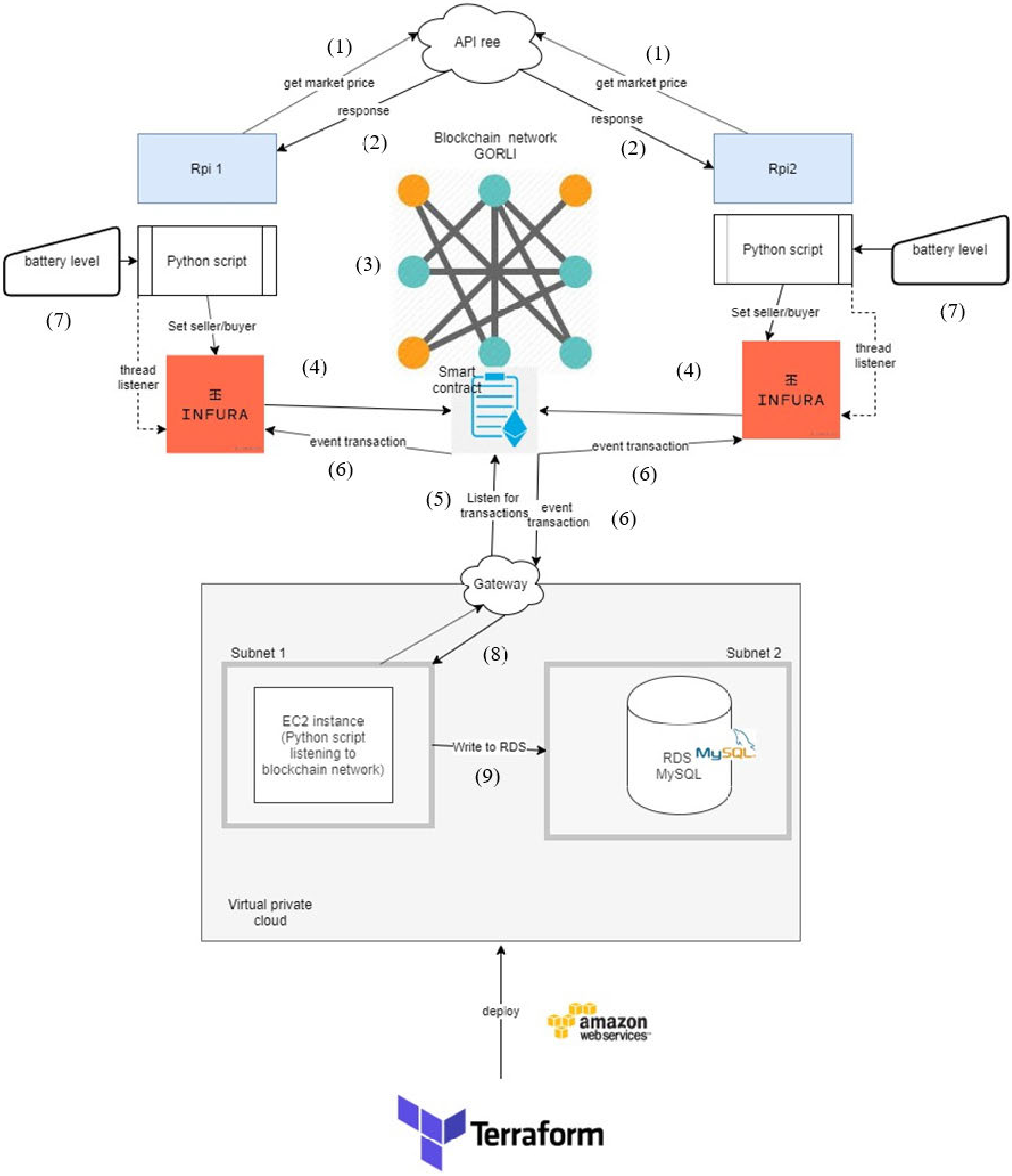


FIGURE 6. Deployment of the prototypes created and their relationship with all the other parts of the system.

taken to do so have finished successfully. Thus, if the activities that were to be studied as mentioned at the beginning of the manuscript are taken into account again (prices in real time from a TSO, smart contract deployment among peers, trade operations and information storage in a blockchain) it can be considered as proven that a Constrained Hardware Device can be used as an effective solution for transparent energy trading. Cloud computing-enhanced CHDs can therefore be integrated onto devices capable of storing energy to make them reliable enough to trade electricity automatically according to the information regarding prices that can be obtained from a TSO.

V. CONCLUSION AND FUTURE WORKS

Regarding the work that has been done in the conception, design, implementation and testing of the blockchain-based distributed system that was shown there are several conclusions that have been inferred from all these activities.

To begin with, the usability of constrained devices to guarantee trading operations between peer-to-peer entities has been proven. As long as they are capable of having software components installed on them with low requirements of storage or RAM memory, a low capability device with the performance of a Raspberry Pi 3B+ is capable of engaging prosumers in buying and selling electricity in a blockchain-based environment. While the operations that have been described here are shown in the relative controlled environment of a laboratory, it can be said that the CHD conceived, designed, developed and tested here has fulfilled its initial purpose of enabling smart contract development, deployment and execution for any party that has electricity to trade. The fact that the CHD can be extended to any other environment where there are equivalent goods or services that can be traded makes it even more appealing for its potential usage in other IoT or Cyber-Physical System environments.

In addition to that, coexistence among several market infrastructures that are built differently but can serve a similar purpose has been shown as well. In the demonstrator that has been set that makes use of the two prototypes that have been created, it shown how transactions can take place under distributed blockchain characteristics (smart contracts, information shared throughout the nodes of the system) and retrieve information from a more centralized-like entity that is a major participant in the power grid as well (in this case, a Transmission System Operation). It is our idea that markets can change to more decentralized procedures to work, and such procedures are desirable to happen, but such a change will take place gradually, and for a prolonged amount of time both ways of trading energy will take place at once.

Furthermore, the criteria used to measure the capabilities of the proposed model according to what has been introduced in section IV (having a CHD obtaining TSO data in real time, deploying smart contracts on a peer-to-peer basis, enabling trade operations and storing information in a blockchain) have been fulfilled and proven as valid. As it was mentioned before, the CHD that has been built for this manuscript can indeed be used to be plugged to a battery or any other device capable of storing electricity for them to be sold, so it can be deemed as a usable device.

Furthermore, it has also been proven that a significant number of software and hardware tools and components can be integrated in a constrained device built for the purpose of trading with electricity. If there is a common agreement on how to interface each of the components, integration of hardware and software that happens only too often on the Internet of Things or in Cyber-Physical Systems developments can be done in a successful manner.

Finally, if the contributions that were mentioned in the first section of the manuscript are taken into account, it becomes clear that the CHD implementation that has been built in this manuscript has fulfilled them. This Raspberry Pi-based CHD a) can deploy smart contracts written in Solidity despite its constrained nature, b) makes use of a lightweight consensus algorithm (Proof-Of-Authority) and can store backup information in cloud facilities, c) can retrieve information from a TSO in real time with the only limit of the REST interface that is provided and d) due to its usage of blockchain and smart contracts, it effectively removes the intermediaries for utility services.

There are other aspects related to how realistic the deployment of CHDs would be in terms of usability and performance. While Raspberry Pi devices have been used as smart meters with success [41][42], they may be prone to reliability issues if strained after a long timespan. Nevertheless, since it is essentially used to encase all the software developed (blockchain, Infura) and hardware interfaces (especially the network interface) in the same location, any other device capable enough and with the same number of minimum interfaces can be used instead. Network links towards the cloud computing infrastructure have to be considered as well: too many links might result in congestion and/or collapse of the system. To prevent this, gathering of prosumer producers can be done, either via an aggregator or by creating a Virtual Power Plant (VPP) among all the interested parties. One additional positive aspect of building a CHD as the one shown in this manuscript would be its inexpensive nature; the components required to have it running cost no more than $70 in total if the device, Light Emitting Diodes, Raspberry Pi and any other auxiliary hardware (relays for electric flow control) are included in the budget. However, work conception, development and testing are major and should be introduced in the final retail price of the CHD that was manufactured. Limitations on the capabilities of Raspberry Pi devices should not come as a deal-breaker for the CHD that is put forward in this manuscript: it has been tested as a success not only as a smart meter, but also as a key component in an optophone [43] or an offline media server [44] among other developments. In the end, the device that is being put forward in the paper performs a limited number of functionalities: a) connection to REST API to obtain electricity prices for a specific timeslot, b) deployment and execution of smart contract onto a distributed blockchain and c) storage of data both in the blockchain and in a cloud-based infrastructure. The number of times these actions are executed for 24 hours comes as reasonably limited, as electricity tends to be priced in different daily timespans of several hours of duration; an example of this price distribution can be seen in [45], where it is shown how there tend to be different energy prices depending on the time of the day. Arguably, trade operations to be carried out will not have to be that frequent for a regular prosumer, so the computational power that has been shown by the Raspberry Pi can be deemed as enough. Furthermore, considering the Raspberry Pi 3B+ technical specifications as described in [25], several similar low-cost computers (i.e., BeagleBone Black [46]) can be used for the same purpose of building a CHD that can be connected to a device capable of storing and using electricity.

There are several future lines of work that can be set as a continuation to the efforts done. A more sophisticated way to obtain energy forecasting features might be able to be run in the CHD, or at least part of this procedure (generally speaking, any kind of procedure that requires the most significant computational power) could be outsourced to the cloud, so algorithm training takes place with more capable infrastructure. Furthermore, a fully operational network with more than two devices can be deployed in a facility so that more complex trading operations with several participants in the blockchain-based distributed system can take place. Lastly, rather than using an extremely useful yet generic device such as a Raspberry Pi, other pieces of hardware can be tailored for the needs of the software to be installed. In that case, performance of the CHD might be boosted.

All the achievements done related to this piece of research are being openly offered to the research and development communities as a GitHub repository that can be freely accessed at [47].

REFERENCES

[1] *Quarterly Report on European Electricity Markets With Focus on Energy Storage and 2019 Wholesale Prices*, Market Observatory for Energy, DG Energy, European Commission, Brussels, Belgium, 4th Quart., 2019, vol. 12, no. 4.

[2] M. Sabounchi and J. Wei, "Towards resilient networked microgrids: Blockchain-enabled peer-to-peer electricity trading mechanism," in *Proc. IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2017, pp. 1–5, doi: 10.1109/EI2.2017.8245449.

[3] J. Rodríguez-Molina, B. Martínez, S. Bilbao, and T. Martín-Wanton, "Maritime data transfer protocol (MDTP): A proposal for a data transmission protocol in resource-constrained underwater environments involving cyber-physical systems," *Sensors*, vol. 17, p. 1330, Jun. 2017, doi: 10.3390/s17061330.

[4] J. Rodríguez-Molina, J.-F. Martínez, and P. Castillejo, "A study on applicability of distributed energy generation, storage and consumption within small scale facilities," *Energies*, vol. 9, no. 9, p. 745, Sep. 2016, doi: 10.3390/en9090745.

[5] Power Ledger Website. *The Operating System for New Energy Markets, Power Ledger is a Blockchain Enabled Software Platform for Trading Renewable Energy and Environmental Commodities*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.powerledger.io/

[6] WePower Website. *The Easiest Way to Buy Green Energy Directly From Producers and Reach Your Energy Sustainability Goals*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.wepower.com

[7] A. S. Musleh, G. Yao, and S. M. Muyeen, "Blockchain applications in smart grid–review and frameworks," *IEEE Access*, vol. 7, pp. 86746–86757, 2019, doi: 10.1109/ACCESS.2019.2920682.

[8] *Red Eléctrica de España Main Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.ree.es/en.

[9] A. Laszka, S. Eisele, A. Dubey, G. Karsai, and K. Kvaternik, "TRANSAX: A blockchain-based decentralized forward-trading energy exchanged for transactive microgrids," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Singapore, Dec. 2018, pp. 918–927, doi: 10.1109/PADSW.2018.8645001.

[10] J. Rodriguez-Molina and D. M. Kammen, "Middleware architectures for the smart grid: A survey on the state-of-the-art, taxonomy and main open issues," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2992–3033, 4th Quart., 2018, doi: 10.1109/COMST.2018.2846284.

[11] P. Siano, G. De Marco, A. Rolan, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3454–3466, Sep. 2019, doi: 10.1109/JSYST.2019.2903172.

[12] K. Shuaib, J. A. Abdella, F. Sallabi, and M. Abdel-Hafez, "Using blockchains to secure distributed energy exchange," in *Proc. 5th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2018, pp. 622–627, doi: 10.1109/CoDIT.2018.8394815.

[13] S. Chakraborty, T. Baarslag, and M. Kaisers, "Energy contract settlements through automated negotiation in residential cooperatives," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2018, pp. 1–6, doi: 10.1109/SmartGridComm.2018.8587537.

[14] A. S. Nurgaliev, I. S. Vershinin, and R. S. Minyazev, "Concept and implementation of blockchain demonstration unit within the smart grid," in *Proc. Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, Mar. 2019, pp. 1–6, doi: 10.1109/ICIEAM.2019.8742927.

[15] N. Lasla, M. Al-Ammari, M. Abdallah, and M. Younis, "Blockchain based trading platform for electric vehicle charging in smart cities," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 80–92, 2020, doi: 10.1109/OJITS.2020.3004870.

[16] X. Wang, W. Yang, S. Noor, C. Chen, M. Guo, and K. H. van Dam, "Blockchain-based smart contract for energy demand management," *Energy Procedia*, vol. 158, pp. 2719–2724, Feb. 2019, doi: 10.1016/j.egypro.2019.02.028.

[17] J. Lin, M. Pipattanasomporn, and S. Rahman, "Comparative analysis of auction mechanisms and bidding strategies for P2P solar transactive energy markets," *Appl. Energy*, vol. 255, Dec. 2019, Art. no. 113687, doi: 10.1016/j.apenergy.2019.113687.

[18] A. Amato, B. Di Martino, M. Scialdone, and S. Venticinque, "Towards a SLA for collaborating smart solar-powered micro-grids," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2014, pp. 592–597, doi: 10.1109/INCoS.2014.22.

[19] X. Tan, Y. Wu, and D. H. K. Tsang, "Optimal energy trading with battery energy storage under dynamic pricing," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2014, pp. 722–727, doi: 10.1109/SmartGridComm.2014.7007733.

[20] C. Long, J. Wu, Y. Zhou, and N. Jenkins, "Aggregated battery control for peer-to-peer energy sharing in a community microgrid with PV battery systems," *Energy Procedia*, vol. 145, pp. 522–527, Jul. 2018, doi: 10.1016/j.egypro.2018.04.076.

[21] H. Taylor and L. W. Hruska, "Standard smart batteries for consumer applications," in *Proc. 10th Annu. Battery Conf. Appl. Adv.*, Jan. 1995, p. 183, doi: 10.1109/BCAA.1995.398541.

[22] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Doctoral Diss., Univ. California, Irvine, CA, USA, 2000. Accessed: Oct. 31, 2020. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

[23] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85727–85745, 2019, doi: 10.1109/ACCESS.2019.2925010.

[24] M. Sigwart, M. Borkowski, M. Peise, S. Schulte, and S. Tai, "A secure and extensible blockchain-based data provenance framework for the Internet of Things," *Pers. Ubiquitous Comput.*, Jun. 2020, pp. 1–15, doi: 10.1007/s00779-020-01417-z.

[25] *Raspberry Pi 3 Model B+ Datasheet* Accessed: Oct. 31, 2020. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf

[26] *Görli Testnet Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://goerli.net/

[27] X. Liu, G. Zhao, X. Wang, Y. Lin, Z. Zhou, H. Tang, and B. Chen, "MDP-based quantitative analysis framework for proof of authority," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2019, pp. 227–236, doi: 10.1109/CyberC.2019.00046.

[28] *Infura Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://infura.io/

[29] *REData API Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.ree.es/es/apidatos

[30] *Amazon Web Service Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://aws.amazon.com/free/?nc1=h_ls&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc

[31] *Terraform Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.terraform.io/

[32] J. C. Patni, S. Banerjee, and D. Tiwari, "Infrastructure as a code (IaC) to software defined infrastructure using Azure Resource Manager (ARM)," in *Proc. Int. Conf. Comput. Perform. Eval. (ComPE)*, Shillong, India, 2020, pp. 575–578, doi: 10.1109/ComPE49325.2020.9200030.

[33] *GitHub Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://github.com/

[34] Remix Website. *Remix—Ethereum IDE & Community*. Accessed: Oct. 31, 2020. [Online]. Available: https://remix-project.org/

[35] Solidity Website. *Solidity—Solidity 0.7.4 documentation*. Accessed: Oct. 31, 2020. [Online]. Available: https://solidity.readthedocs.io/en/v0.7.4/

[36] *Truffle Suite Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.trufflesuite.com/

[37] *Metamask Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://metamask.io/

[38] Postman Website. *Postman | the Collaboration Platform for API Development*. Accessed: Oct. 31, 2020. [Online]. Available: https://www.postman.com/

[39] *Etherscan Website*. Accessed: Oct. 31, 2020. [Online]. Available: https://etherscan.io/

[40] S. Zhai, Y. Yang, J. Li, C. Qiu, and J. Zhao, "Research on the application of cryptography on the blockchain," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 032077, doi: 10.1088/1742-6596/1168/3/032077.

[41] P. A. Chandra, G. M. Vamsi, Y. S. Manoj, and G. I. Mary, "Automated energy meter using WiFi enabled raspberry Pi," in *Proc. IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2016, pp. 1992–1994, doi: 10.1109/RTEICT.2016.7808186.

[42] R. de Diego, J.-F. Martínez, J. Rodríguez-Molina, and A. Cuerva, "A semantic middleware architecture focused on data and heterogeneity management within the smart grid," *Energies*, vol. 7, no. 9, pp. 5953–5994, Sep. 2014, doi: 10.3390/en7095953.

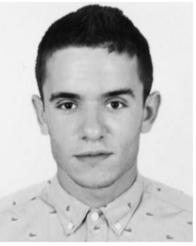
[43] Y. Li, J. Cheng, and X. Wang, "An optophone based on raspberry pi and Android wireless communication," in *Proc. IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. (AEECA)*, Aug. 2020, pp. 952–956, doi: 10.1109/AEECA49918.2020.9213587.

[44] A. P. Jadhav and V. B. Malode, "Raspberry PI based OFFLINE MEDIA SERVER," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 531–533, doi: 10.1109/ICCMC.2019.8819718.

[45] *Active Energy Invoicing Price*. Accessed: Mar. 31, 2021. [Online]. Available: https://www.esios.ree.es/en/pvpc

[46] *BeagleBone Black System Reference Manual*. Accessed: Mar. 31, 2021. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf

[47] *Final Master Thesis, by Pablo Méndez Royo*. Accessed: Oct. 31, 2020. [Online]. Available: https://github.com/pablomendezroyo/finalMasterThesis



PABLO MÉNDEZ ROYO is currently an Engineer of the Internet of Things. He also works as a Software Engineer in distributed computing developments and holds several blockchain certifications, and has international experience both in industry and education with regards to information and communication technologies. He successfully defended his Master Thesis at the Technical University of Madrid, in 2020, called "Device built with inexpensive hardware for energy storage, trade, and transfer via blockchain." His research interest includes distributed systems, such as software development, blockchain, the IoT, networking, and distributed security.

JESÚS RODRÍGUEZ-MOLINA is currently an Assistant Professor with the Technical University of Madrid. He received his Ph.D. Thesis with honors, in 2017, for contribution to the design, implementation and standardization of semantic middleware architectures for the smart grid. He has performed research activities in Switzerland (ETH), Norway (SINTEF), Colorado (NREL, CU Boulder), and Viena (TU Wien). His research interests comprehend distributed and cyber-physical systems, blockchain, autonomous vehicles, and the smart grid and middleware, and has recently started getting familiar with deep learning.

JUAN GARBAJOSA (Senior Member, IEEE) spent 16 years in industry and Government in different engineering and management positions, ten of which at a start-up at the time, which is currently a large multinational company. In 1997, he joined UPM, as a full-time Professor. He is currently a Professor with the School of Computer Systems (ETS de Sistemas Informáticos), Universidad Politécnica de Madrid (Technical University of Madrid, UPM), Spain. He also holds the position of the deputy vice-rector for quality systems and competitiveness. His research activity has often addressed industry-academy collaboration, having led numerous international projects. His publications include leading journals and conferences. His current research interests include cyber physical systems architecture, agile, and innovation, and more recently collective intelligence. He is a Convenor of ISO/IEC JTC1 SC7 WG20 2001–2015.

PEDRO CASTILLEJO is currently an Assistant Professor with the Technical University of Madrid. He successfully defended his Ph.D. Thesis with honors, in 2015, called "Contribution towards intelligent service management in wearable and ubiquitous devices." He has performed a research at the Norwegian University of Science and Technology, Norway. He has also participated as an Invited Lecturer in different undergraduate, master, and doctoral courses. His current research interests include blockchain and distributed security, network security algorithms, network protocols, semantics, and intermediation software architectures.

• • •