IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 16, 2020, accepted March 26, 2020, date of publication April 2, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985254

Multi-Agent Microgrid Management System for Single-Board Computers: A Case Study on Peer-to-Peer Energy Trading

LUIS GOMES[1], **ZITA A. VALE**[2], **(Senior Member, IEEE),**
AND JUAN MANUEL CORCHADO[3,4,5], **(Member, IEEE)**

[1]GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto (IPP), P-4200-465 Porto, Portugal
[2]Polytechnic of Porto (IPP), P-4200-465 Porto, Portugal
[3]BISITE Research Group, University of Salamanca, 37007 Salamanca, Spain
[4]Air Institute, IoT Digital Innovation Hub, 37188 Salamanca, Spain
[5]Department of Electronics, Information and Communication, Osaka Institute of Technology, Osaka 535–8585, Japan

Corresponding author: Luis Gomes (lufog@isep.ipp.pt)

This work was supported by the MAS-Society Project cofunded by Portugal 2020 Fundo Europeu de Desenvolvimento Regional (FEDER) through PO CI, under Grant UIDB/00760/2020 and Grant SFRH/BD/109248/2015.

ABSTRACT Smart grids concept benefits and leverage distributed management systems while allowing its players to actively participate in the smart grid. This paper merges the concepts of microgrid and transactive energy. The proposed model is tested in an office building with multiple tenants. An agent-based platform, running in single-board computers, for microgrid intelligent management with a peer-to-peer energy transaction model is proposed in this paper. This paper describes the peer-to-peer transaction auction model and the deployment of the platform in an office building. The results regard a one-week period where the use of peer-to-peer transactions is compared with a scenario where no transactions among agents are performed. The results are promising, showing the energy price inside the microgrid dropping for the majority of players/agents. The presented work demonstrates how smart grid players can decrease their energy costs using simple approaches that do not require load shifting, consumption optimization nor the acquisition of new equipment.

INDEX TERMS Local energy auctions, microgrids, peer-to-peer transactions, transactive energy.

I. INTRODUCTION

Power systems are becoming more distributed in what regards the management and the participation of consumers. This enables the emergence of smaller communities such as micro-grids, where local energy management is performed using local energy demand and local renewable energy sources. Depending on its configuration, a microgrid can work connected to the grid or in islanded mode; i.e. disconnected from the main grid.

The new power systems paradigm enables not only the emergence of small communities but also empowers end-users, that can manage their energy resources and actively participate in the smart grid. End-users can take part in demand response programs and energy transactions, which can result in lower energy prices and enables the intensive and widespread use of renewable energy sources.

The use of Transactive Energy (TE) allows buildings to act as active players using a market-driven approach in the scope of smart grids. The GridWise Architecture Council defines TE as "*a system of economic and control mechanisms that allows the dynamic balance of supply and demand across the entire electrical infrastructure using value as a key operational parameter*" [1]. Although multiple definitions do exist, this one is sufficiently open to cope with the current state of the art.

The concept of TE can be applied to any part or component of power systems, from the transmission to the distribution level, including distributed resources and consumers. This concept includes the management of generation and consumption to ensure the required constant balance between them. This balance gain importance in the frame of intensive

---

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

use of renewable-based generation and distributed generation with strong stochastic nature. TE also enables smart homes and buildings to engage in automated market-based trading in a two-way negotiation [2].

Smart devices, from IoT, are being spread in homes and buildings worldwide. Only in 2007, and in the United States alone, 35.9 million of these devices were sold; excluded from this number are smart televisions [3]. It is expected that by 2022, 216.9 million homes worldwide will have at least one smart device [4]. The vast massification of such devices with remote monitoring and control capabilities opens new possibilities for smart homes and energy management systems. Therefore, they should be considered and integrated into energy management systems.

Multi-agent systems (MAS) provide adequate representation and operational support for distributed intelligent environments. Agents can represent physical devices and infrastructures, ranging from single sensors to entire buildings. Some applications of MAS in microgrids can be seen in [5]–[7], where multi-agent-based microgrid management systems are proposed, enabling distributed consensual optimizations [5], distributed cyber-physical models for real-time tasks [6], and islanded microgrid operation [7]. The use of MAS allows the individual agent-representation of each microgrid player, enabling the exchange of data and information among them. This allows the build of distributed intelligent communities (e.g. microgrids) able to compete and/or cooperate to achieve individual and common goals.

The main novelties of this work are the proposal of a decentralized peer-to-peer transaction model and its implementation and deployment in a microgrid. The decentralization of this model allows and incentivizes end-users to compete among them to pursue their personal goals (i.e. minimize costs) and allows the microgrid to pursue its global goal (i.e. incentivize energy transaction among end-users to decrease the need of external energy supply). It also avoids the need for centralized or external players to manage and decide peer-to-peer transactions among end-users. This is enabled by the proposed use of μGIM multi-agent system (MAS), where each microgrid player is represented by an individual agent running in a single-board computer (SBC). The main contribution, of this paper, is the detailed results achieved by the peer-to-peer transaction model executed by μGIM system. This MAS was designed focusing on the end-user needs and allows the management of the end-users facility while interacting with other players. This work demonstrates that an end-user-oriented agent-based energy management system, like μGIM, can also perform player interaction and execute microgrid auctions for peer-to-peer transactions.

The paper presents promising results using an office building divided by four tenants where each one is a prosumer; the building's manager/owner is also represented by an agent. The five agents/players have the liberty to participate in the peer-to-peer auctions where they can sell energy lots and make bids to energy lots. A distributed open auction model is tested, without any centralized energy management system.

After this first introductory section, the paper presents, in section II, some of the most successful microgrid implementations where transactive energy is a reality. Section III presents the μGIM platform and section IV presents the proposed transactive energy model. The microgrid deployment is presented in section V and the main results can be seen in section VI. Finally, the main conclusions are presented in section VII.

II. MICROGRIDS AND TRANSACTIVE ENERGY

This section is divided into three topics: microgrids, MAS for microgrids, and transactive energy. The first topic presents microgrid deployments of success stories. In the second topic, two open-source software platforms for microgrids and energy management systems are presented. And the third topic presents projects and real deployments where transactive energy can be found.

A. MICROGRIDS

Universities have a good potential to adopt microgrids on their campus for research and energy cost reduction. Microgrids in University campus are therefore relatively common. Two success stories are the New York University [8] and the University of California in San Diego [9] where savings range from 5 to 8 million US dollars, in New York, and 8 million US dollars, in San Diego.

The capability, of microgrids, to work in islanded mode (i.e. disconnected from the main grid) called the attention of military bases, allowing them to be independent of the main grid. Therefore, some pilots were implemented in military bases, such as Camp Pendleton Microgrid with 530 kW of photovoltaic peak generation [10], and Fort Carson Microgrid with an outstanding 2 MW of photovoltaic peak generation [11].

The Brooklyn microgrid is also a very interesting pilot that is integrated into the community [12]. This pilot enables peer-to-peer transactions at the end-user level, where end-users can specify the price that they are willing to pay and the type of energy source that they are willing to acquire; e.g. pay more for solar generation. The possibility to specify prices for each energy source, allows end-users to gain control over the supplied energy. This microgrid integrates prosumers and consumers. Blockchain is used, in the microgrid, to build a complete distributed system among end-users, without the need of a centralized player. End-users use a mobile application to have control over the energy that they want to sell and the energy they want to buy, and how the operations are made. The Brooklyn microgrid integrates around fifty end-users using Transactive Grid elements (TAG-e meters) that enable virtual peer-to-peer transactions using the utility grid [13].

B. MULTI-AGENT SYSTEMS FOR MICROGRIDS

The management of microgrids can be made using VOLTTRON. Developed at Pacific Northwest National Laboratorial (PNNL), in the United States, VOLTTRON was released as open-source in 2014 [14], with version 4.0 being released at the end of 2016. VOLTTRON is an agent-based solution running in python that can run in SBC, such as Raspberry Pi 3 Model B and Raspberry Pi Zero. For resource integration, VOLTTRON uses its driver framework where protocols such as BACnet and Modbus can be used. VOLTTRON enables energy transactions and it is compatible with the Automated Demand Response (OpenADR) standard [15]. However, it is not FIPA (Foundation for Intelligent Physical Agents) compliant. PNNL has a campus building where VOLTTRON is deployed and where physical experiments are conducted for market-based transactive controls for heating, ventilation, and air-conditioning (HVAC) systems [16].

The resilient Information Architecture for the Smart Grid (RIAPS) framework can also be used for microgrid management, providing run-time and design-time environments for smart grid applications [17]. RIAPS uses python for soft real-time and C++ for hard real-time, and it can be run in SBC such as Beaglebone Black Board with Linux operating system. This framework is a multi-thread software that communicates between threads using messages. Although it is currently under development, RIAPS must be taken into consideration for new implementations of smart grids and microgrids. Initial case studies demonstrate RIAPS capability for transactive energy applications [18] and distributed control for microgrid synchronization [19].

Besides VOLTTRON and RIAPS, other options are available and smaller and direct approaches are proposed. In Florida International University a smart grid testbed using tailor-made agents running in Beaglebone Black SBC was developed [20]. Because these solutions try to solve very specific issues and do not allow the scalability that is enabled by complete systems, such as RIAPS and VOLTTRON, they were excluded from this section. Only complete and robust multi-agent systems for microgrids were presented.

C. TRANSACTIVE ENERGY

Transactive energy (TE) is a relatively new concept that enables end-users to have more active participation in the smart grid. Being a broader and embracing concept, it enables multiple possible participations where end-user can be part of.

Several methodologies and implementations of TE, using transactive control, have been developed [21]. The "Clean Energy and Transactive Campus Project" in the US [22] and the PowerMatcher in Europe (http://flexible-energy.eu/powermatcher) should be highlighted. Both projects provided significant scientific developments and practical testbed implementations.

This paper proposes a peer-to-peer energy trading model that enables end-users to trade energy among themselves. This enables end-users to have a participating role in the microgrid and enables them to lower energy costs or even generate profits. Some projects proposing similar approaches can be found in [23].

Within microgrids, project PeerEnergyCloud and the already described Brooklyn microgrid enabled peer-to-peer transactions among microgrid players. The PeerEnergyCloud uses a cloud-based approach [24]. Players need to have a cloud connection interface (in this case fiber optic is used), where they can use forecasting services and participate in the peer-to-peer market. A storage service is also provided in the cloud.

A special reference must be done to project T77 in Bangkok, Thailand, led by BCPG and Power Ledger companies [25]. The first trade of T77 happened on the 22nd of August 2018, but the project is still under development. With a total of 400 kW installed photovoltaic capacity, it has a higher generation capacity than consumption demand, enabling energy trading according to the buildings' needs.

Energy peer-to-peer trading can assume several forms and models. The authors would like to point out a successful deployment of peer-to-peer transactions with an original model that revolutionized two unelectrified villages in India. This project was implemented in Rampur and Manpur, India [26]. The end-user in this project can rent solar-items – equipment with batteries charged using the installed solar panels, such as LED bulbs. The rent payment is argued between buyer and seller and where between 27% to 45% of payments were made without using cash. This model shows the real potential of microgrids and peer-to-peer trading to provide a new life to unelectrified villages.

In research publications, peer-to-peer transactions can be found in several simulations works, such as in [27] where a peer-to-peer non-cooperative auction model is used by applying Nash equilibrium [28] and ECO-Trade algorithm is proposed to coordinate peer-to-peer trading and demand-side management, and in [29] where a price-adjustment process is applied in peer-to-peer auctions. However, they lack the deployment in physical buildings. This paper presents a peer-to-peer transaction model implemented and deployed in a physical microgrid office building.

III. THE μGIM SOLUTION

In this paper, it is proposed the Microgrid Intelligent Management (μGIM) platform for peer-to-peer transactions among microgrid players. This novel agent-based platform enables the management of the building, and of its resources, and the transaction of energy among players. Each μGIM agent runs in an SBC. The main advantages of this system, compared to RIAPS and VOLTTRON, is that μGIM is centered in the end-user rather than the microgrid. The μGIM agents are capable to run in standalone mode (i.e. disconnected from a multi-agent system), providing energy management methodologies to the end-user [30]. Being centered in the end-user, the μGIM was designed and built from the end-user to the grid, while RIAPS and VOLTTRON are grid oriented. All three systems can be executed in single-board computers, but the only μGIM supports energy strategies developed in several languages. An energy strategy is an executable class that performs a task, such as a forecast, resource optimization, and demand response event reply. In the μGIM MAS, each

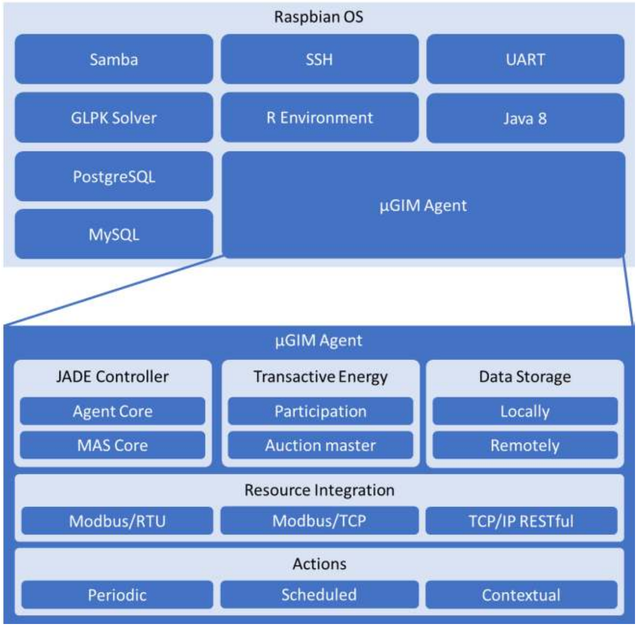


FIGURE 1. **μ GIM overall architecture.**

microgrid player is represented by an agent running in a dedicated SBC.

Considering the seven-layers agent-based control architecture for smart grids proposed in [31], μGIM agents will actuate, depending on their configuration, in the following layers: prosumer agents, DER agents, and microgrid agents. In the μGIM platform, the layer component level and load level agents are not considered; loads and resources will have a direct connection with μGIM agent using the resource integration module that will be presented in this section.

According to Wooldridge [32], agents in a multi-agent system must have the following abilities: reactivity, pro-activeness, and social. In μGIM, all agents have these three abilities. They can react to changes in their environments, they are responsible for monitoring and controlling their energy resources and they can detect changes and react using energy management methodologies, as seen in [30]. The reactivity ability is also used during auctions to bid against the lot's highest bid. In the proposed peer-to-peer transaction auction model, agents show pro-activeness by trying to pursue their goals (e.g. selling the surplus energy or buying the deficit energy). The social ability is intrinsically linked to the participation in the auctions, where agents exchange lots and bids.

The μGIM platform runs in Raspberry Pi boards using the operating system (OS) Raspbian; with or without a desktop interface. Other SBC boards are being tested, such as Orange Pi (Ubuntu), Cubie Truck (Linaro OS), and BeagleBone Black (Ubuntu).

The μGIM platform is used to empower microgrid players with transactive energy capabilities. The platform can monitor and control resources, perform energy forecasting and interact with neighbors to sell or buy energy. Figure 1 shows the architecture of the SBC used in μGIM. This architecture presents the needed modules for transactive energy; a complete detailed architecture of μGIM agents can be found in [33].

Raspbian OS is used with Samba software and Secure Shell (SSH) to interact with the agent. Although Raspbian OS has a version with a desktop graphical interface, the used OS does not provide such an interface. Samba software is used to exchange files and SSH protocol is used to configure and operate the μGIM agent. The Universal Asynchronous Receiver-Transmitter (UART) in the SBC can also be used to enable the direct integration of the SBC into an RS485 network where Modbus/RTU protocol is used.

The μGIM agent is developed in Java, version 8. For energy management proposes, GLPK solver and R environment are installed in Raspbian OS. This allows the μGIM agent to use solvers for large-scale linear programming and mixed-integer programming, as well as the available packages of R that can be used for optimization, aggregation, categorization, and forecast.

The SBC used has two available databases: PostgreSQL and MySQL. Both can be used by μGIM, depending on the agent configuration, but they cannot be used simultaneously. Although they are both available in the μGIM agent, PostgreSQL is used in this work because it has an open-source driver, while MySQL driver is under GPL license.

The μGIM agent, here presented in a very concise way, enables the resource integration using three protocols, that cover the majority of electrical resources that can be found in today's buildings: Modbus/RTU, Modbus/TCP, and TCP/IP RESTful. The first two are variations of Modbus protocol, where the first operates over an RS485 network and the second operates over a TCP/IP network. Modbus is largely used in energy-related equipment and covers a significant part of energy analyzers available in today's market. The TCP/IP RESTful is not, in fact, a protocol, this block represents the integration of resources with available RESTful Application Programming Interface (API), meaning that they follow the REST software architecture.

An action mechanism is implemented in μGIM agents, allowing the trigger of actions. An action is an interface class in Java that can do almost anything. An action can be periodic, meaning that it will be executed in a periodic time, scheduled, meaning that it will only be executed once, or contextual, meaning that it will be executed in a specific scenario/context. For instance, a contextual action can be an alarm that detects electrical loads turned on during the night. The alarm mechanism is a Java class that extends the class Thread and runs an infinity loop every second. This class monitors the list of available actions and executes the ones that need to be executed.

The μGIM uses JADE framework to enable the use of MAS. Depending on its configuration, any μGIM agent can be the MAS core, meaning that it will run the main container of JADE. Because the μGIM platform is distributed and does

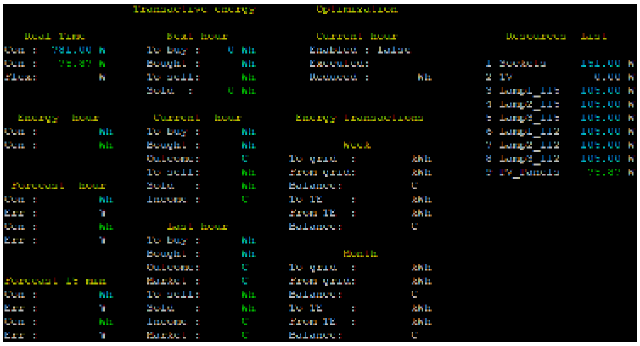


FIGURE 2. μ**GIM command-line interface.**

not use a centralized server, the main container of JADE runs in one of the agent's SBC. Each SBC running μGIM will execute its own container in JADE and use the local network to connect with the main container. Currently, all μGIM agents must be in the same local network, physically or using a virtual private network (VPN), to allow connections with the others. Agents in different local networks are not recommended in μGIM because security measures are currently not developed.

The MAS enabled by μGIM agents has the global objective of creating a microgrid where agents are incentivized to participate and pursue their individual goals in a stable microgrid. To incentivize transactive energy in the microgrid, managed by the μGIM platform, it will be made available peer-to-peer transaction auctions where agents are free to participate to try to reduce their energy costs.

The μGIM platform was already tested regarding its abilities to monitor multiple facilities [33] and to execute a demand-side management algorithm in an office [30]. In this paper, μGIM agents are used to testing the proposed peer-to-peer energy transaction model.

In μGIM platform, each agent can participate in peer-to-peer transaction auctions. To enable the use of auctions, the MAS demands the definition of an auction synchronizer than synchronizes auctions among agents. Auctions are executed by every seller; the auction synchronizer will guarantee that sellers do not start auctions simultaneously. The auction synchronizer role is set to a common agent in μGIM platform. In μGIM, all agents are equal in their code, architecture, and structure.

Figure 2 shows the skeleton of the μGIM graphical interface available in each agent. This interface uses the command line of Raspbian OS to show the overall data.

IV. PEER-TO-PEER ENERGY TRADING MODEL

The peer-to-peer energy trading proposed in this paper assumes that all the microgrid players can buy or sell energy in a given period t. However, they cannot buy and sell within the same period. The amount to sell/buy is supported by energy forecasting algorithms for generation and consumption. All agents internally forecast their energy using local forecast algorithms, running in the SBC. The difference between the forecasted consumption and generation is put to sale or to purchase.

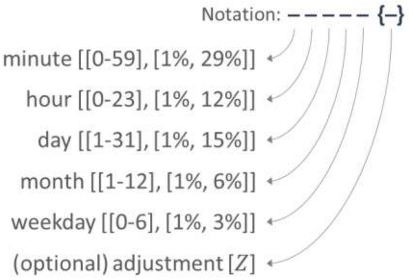


FIGURE 3. μ**GIM schedule notation.**

For this implementation, focused on peer-to-peer transactions, it was used eight forecast algorithms: three baselines according to the last ten days and adjusted to the last periods, two weighted arithmetic average forecasts using the last periods, and three support-vector machines (SVM) algorithms using past periods and/or days. The SVM algorithm use R Project language. These eight algorithms were already available in μGIM agents and they are executed every hour. For the peer-to-peer transaction model, the forecast with the lower error is used. The error is calculated using the mean absolute percentage error (MAPE) of last week. Every Monday a forecast algorithm for consumption and a forecast algorithm for generation are chosen. Because MAPE cannot handle real values equal to zero, generation periods without generation are not considered.

Internally, forecast algorithms are considered actions. In the μGIM platform, agent actions can be set using a similar notation as the Linux crontab expressions. The notation can be seen in Figure 3, where the first five parameters are similar to crontab. The percentual symbol indicates multiples of the specified number; e.g. the notation $10\% \ast \ast \ast\ast$ will run the action every 10 minutes at $\{0, 10, 20, 30, 40, 50\}$ minutes. The last parameter does not exist in crontab notation and it is an adjustment parameter, optional in μGIM. This parameter defines a value in minutes that will be added to the end of notation. By using the previous example, if we added an adjustment of -2 minutes the notation will be $10\% \ast \ast \ast\ast$ -2 and it will be triggered every 10 minutes less 2 minutes at $\{58, 8, 18, 28, 38, 48\}$ minutes.

Each agent is configured with four forecasting actions while the auction synchronizer agent has two additional actions dedicated to the request of available sellers and the start of the auctions:

- $18 \ast \ast \ast\ast$ – forecast hour-ahead energy consumption;
- $18 \ast \ast \ast\ast$ – forecast hour-ahead energy generation;
- $15\% \ast \ast \ast\ast$ -5 – forecast next 15 minutes energy consumption;
- $15\% \ast \ast \ast\ast$ -5 – forecast next 15 minutes energy generation;
- $30 \ast \ast \ast\ast$ – request available sellers; configured only in the auction synchronizer and requested to all producers and prosumers;

- 35 $* * **$ – synchronization among sellers to start auctions; configured only in the auction synchronizer.

The forecasting for the next 15 minutes is executed at $\{10, 25, 40, 55\}$ minutes and it will not be directly used for transactive energy. The hour-ahead forecast will be executed each hour at *hh*:18 minutes (where $hh \in [0,23]$). Energy consumption and energy generation forecast are executed simultaneously using multi-threads. The result determines the amount of energy that the agent will try to sell or buy in the following auctions.

In the proposed peer-to-peer trading model, agents/players will not cooperate, they will be competitive agents trying to reach their personal goals. In this model, μGIM enables the use of four types of auctions that can be used for peer-to-peer trading:

- English – this is an ascending-bid type of auction that enables all participants to bid over the price of the lot, provoking an ascending price scale over time, the price must overpass the last bid and the item is sold when the auctioneer stops receiving bids;
- Dutch – this is a descending-bid type of auction that initiates at a high price and will slowly decrease over time, it can have one or multiple bidders depending on the lot auctioned (e.g. a lot with multiple items), the bidders are ordered by descending price and this indicates the priority of each bidder, and where the bidder with higher priority is the first to select the items in the lot;
- Blind – this is a first-price sealed-bid auction where the lot is known by participants and where each participant can make a unique individual and sealed bid, then the auctioneer opens the sealed bids and the highest bidder wins the lot and pays the presented bid;
- Vickrey – is a second-price sealed-bid auction similar to the blind auction type, but where the highest bidder pays the second-highest bid and not his/her bid.

An auction is considered as being the auction of a unique lot. A set of lots/auctions is considered an auction catalogue. Therefore, there will be one auction catalogue per hour, where several lots can be presented and auctioned. A lot is considered an amount of energy to trade in the next hour. All lots regard energy to be traded in the next hour-period.

The timing of all auctions is parameterized in agents. In our case study, the English auction type is used. The time that the auctioneer (i.e. seller) waits for new bids before close/selling the lot is parameterized in the configuration file.

In our model, all agents are participants in the auctions. However, depending on their forecast, they can be sellers or bidders. It is not possible to play both roles in the same period t. In our model, a lot is considered as an amount of energy (Wh) that agents want to sell in the auction. The bids will be taken in EUR/kWh independently of the size of the energy lot. The price that will be paid is the relation between the bid, in EUR/kWh, and the energy lot size.

Agents are free to choose to participate, or not, in the hourly auctions catalogue according to their individual needs. There is no centralized optimization mechanism for peer-to-peer transactions; agents are independent participants that compete with each other. The proposed model is distributed, and only individual agent goals are pursuit, so there is no need for a central agent to optimize the amount of energy that must be transacted in the microgrid.

Figure 4 shows the sequence diagram of the μGIM platform, where three μGIM agents are represented: the agent responsible for the MAS Core where JADE main container and directory facilitator are running, the auction synchronizer agent that is responsible for transactive energy between players, and a generic prosumer that represents all the platform agents.

Internally, to enable peer-to-peer trading in our model, three recurrent routines are needed (Figure 4). The first routine updates the agents list every minute, by using the services of JADE Directory Facilitator agent. The second recurrent routine allows the hourly notification of market prices, from the auction synchronizer to all other agents. The last routine is the execution of forecasting algorithms, starting every hour at hh:18.

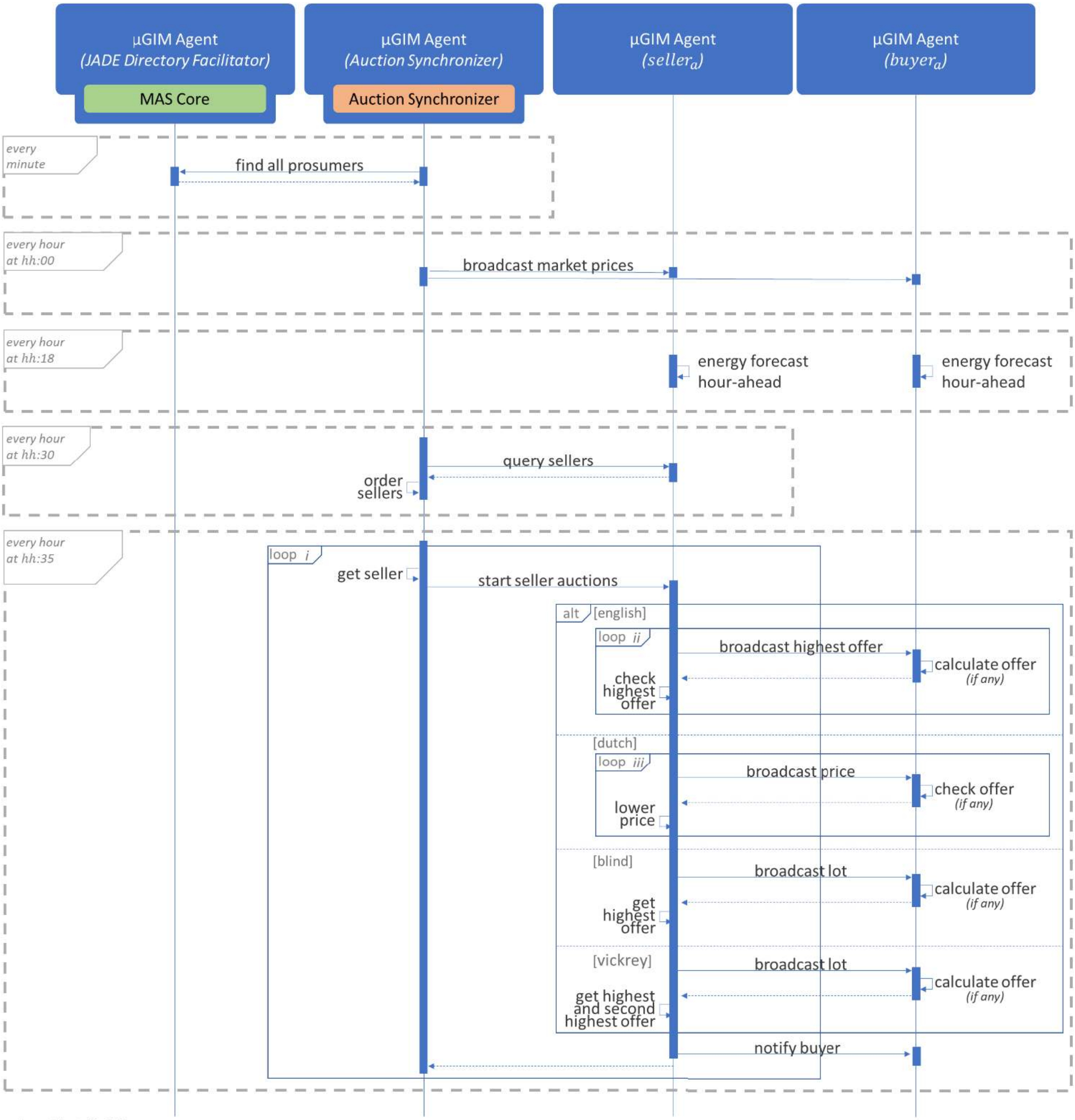
The auction synchronizer is responsible for the synchronization of transactive energy auctions. The auction synchronizer issues two self-triggering events that are configured in μGIM as actions that start the request of sellers and the start of auctions. These self-triggering events send messages to other agents that use them as event-triggering events to reply. At hh:30, the auction synchronizer agent requests all available sellers. At hh:35, the auction synchronizer starts the auctions using a FIFO (first in, first out) methodology, where sellers are server in the order they were presented announced. Auctions stop when no more sellers are waiting.

Lots are auctioned every hour, and each hour has its auction catalogue, aggregating all lots of all sellers. In each auction catalogue, the total amount of energy (Wh) to be traded in peer-to-peer transactions is calculated according to equation (1).

$$ME_{ac} = \min\left(\sum_{a=1}^{m} E^{s}_{a_{ac}}, \sum_{a=1}^{m} E^{b}_{a_{ac}}\right) \tag{1}$$

where m represents the number of agents participating in the peer-to-peer transaction auctions, $E^{s}_{a_{ac}}$ is the amount of energy that agent a has to sale in auctions catalogue ac, and $E^{b}_{a_{ac}}$ is the amount of energy that agent a wants to buy in auctions catalogue ac. An auctions catalogue, ac, represents a set of auctions that are executed within the same hour. In a day, there are 24 auctions catalogues.

The economic trading amount in one hour is represented by $\sum_{i=1}^{n} L^{hb}_{i_{ac}}$, where n is the number of available lots and $L^{hb}_{i_{ac}}$ is the highest bid offered for lot i in auctions catalogue ac. The economic trading amount, equation (2), is a value between zero – in the case where there are no buyers or sellers – and the total amount of energy traded in the auctions catalogue, ME_{ac}, multiplied by the maximum offer, $MO_{a_{ac}}$, available



where $hh \in [0, 23]$
Loops conditions:
i. while there are sellers
ii. while the selling time, of 15 seconds, is not reached without having an higher bid
iii. while no offer is presented, or the price reaches values below minimum price

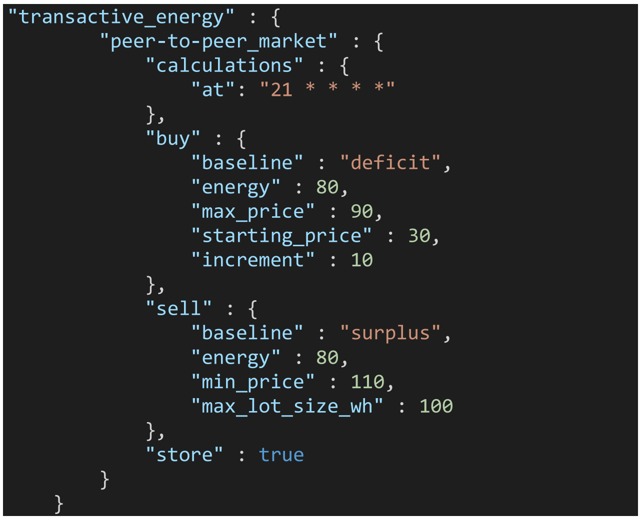
FIGURE 4. Sequence diagram for transactive energy.

among microgrid agents.

$$0 \leq \sum_{i=1}^{n} L_{i_{ac}}^{hb} \leq ME_{ac} \times \max_{0\leq a\leq m} (MO_{a_{ac}}) \qquad (2)$$

where $\sum_{i=1}^{n} L_{i_{ac}}^{hb}$ indicates the sum of all economic transactions, summing all the highest bids of available lots, and m represents the number of agents in the microgrid. Lots that did not receive bids will have a $L_{i_{ac}}^{hb} = 0$.

Figure 5 shows the agent configuration file, using JavaScript Object Notation (JSON) format. Each agent has its configuration file. The ''calculations.at'' key defines a scheduled action that calculates the energy to be sold or bought. This action is scheduled for 3 minutes after the execution



```
"transactive_energy" : {
        "peer-to-peer_market" : {
            "calculations" : {
                "at": "21 * * * *"
            },
            "buy" : {
                "baseline" : "deficit",
                "energy" : 80,
                "max_price" : 90,
                "starting_price" : 30,
                "increment" : 10
            },
            "sell" : {
                "baseline" : "surplus",
                "energy" : 80,
                "min_price" : 110,
                "max_lot_size_wh" : 100
            },
            "store" : true
        }
    }
```

FIGURE 5. **μGIM agent's transactive energy configuration.**

of the forecasting algorithms. The "store" key set with a true value, gives the information that transactions need to be stored in the database.

The configuration "baseline" for buying energy can have the following values: "deficit", "all", "infinite" or "none". The "deficit" value is the deficit between consumption and generation. The "all" value indicates that the agent will try to buy the total amount of consumption. The "infinite" value is for test only and will force the agent to bid to every lot. The "none" value indicates that the agent will not make any bid.

The "buy" key defines how many and how the agent will buy energy in the transactive energy auctions. The energy, "max_price", "starting_price" and "increment" keys are percentual numbers. For example, in Figure 5 the agent will buy 110% of the deficit between generation and consumption. The "max_price" key defines the maximum price that the agent will offer for a lot. In this example, that agent will offer until 90% of the market price for energy purchase from the main grid. The "starting_price" defines the minimum price that the agent will offer for a lot; 40% of the market price for energy purchase. The "increment" key represents the percentual increment that the agent will apply to the last highest bid.

The "baseline" in "sell" key can be set to: "surplus", "all" or "none". The "surplus" value indicates that the agent will sell the surplus energy between generation and consumption. The "all" value makes the agent to sell any energy generated. The "none" value forces the agent to not sell any energy.

The "energy" key indicates the percentual value of the energy that will be put to auction considering the baseline. In this example, the agent will try to sell 80% of the surplus energy. The "min_price" indicates that the agent will only accept bids higher than 100% of the market price for energy sale – i.e. energy injected to the main grid. The "max_lot_size_wh" indicates the maximum size, in Wh, of each energy lot.

Sellers must divide their lots before starting the auctions. This division is not mandatory but highly recommended. If an agent needs to buy 150 Wh, then it will not bid on lots with more than 150 Wh. If big lots are not divided, they can have no one bidding because they surpass the agent's energy needs/targets. Agent a will put energy lots to sale if equation (3) is higher than 0.

$$E^s_{a_{ac}} = \begin{cases} 0 & \text{if } baseline = none \\ F_{gen_{h+1}} - F_{cons_{h+1}} & \text{if } baseline = surplus \\ F_{gen_{h+1}} & \text{if } baseline = all \end{cases} \tag{3}$$

where $F_{gen_{h+1}}$ represents the hour-ahead generation forecast for auctions catalogue at hour h, and $F_{cons_{h+1}}$ represents the hour-ahead consumption forecast for the same hour. The baseline is given by the "sell.baseline" key in the agent's configuration file. The number of lots that agent a will put to sell on the peer-to-peer auctions in auctions catalogue ac is given by $L^{number}_{a_{ac}}$ of equation (4).

$$L^{number}_{a_{ac}} = \left\lceil \frac{E^s_{a_{ac}}}{ML^s_a} \right\rceil \tag{4}$$

The "max_lot_size_wh" of agent a is expressed as ML^s_a. The minimal price accepted by agent a is equal in all its lots in the same auctions catalogue and it is calculated according to equation (5)

$$L^{min}_{a_{ac}} = mP_a \times \mathrm{M}^S_h \tag{5}$$

where mP_a is the "min_price" in the sell configuration of agent a and M^S_h represents the market price for energy sold to the grid in hour h, the same hour of auctions catalogue ac. For each lot i, in auctions catalogue ac, agents will present offers if the constraint of equation (6) is respected.

$$L^s_{i_{ac}} < E^b_{a_{ac}} - E^{bp}_{a_{ac}} \tag{6}$$

where $L^s_{i_{ac}}$ is lot i size and $E^b_{a_{ac}}$ is agent a energy target to buy and $E^{bp}_{a_{ac}}$ is agent a energy bought, all regarding the same auctions catalogue ac.

If the constraint (5) is respected, then agent a calculates the starting offer (SO_{a_i}) of equation (7), the incremented offer (IO_{a_r}) of equation (8), and the maximum offer (MO_{a_i}) of equation (9). The values of $SO_{a_{i_{ac}}}$ a nd$MO_{a_{i_{ac}}}$ are relative to lot i, while $IO_{a_{r_{i_{ac}}}}$ is relative to a bid request from the auctioneer (i.e. seller).

$$SO_{a_{i_{ac}}} = SP_a \times M^b_h \tag{7}$$

$$IO_{a_{r_{i_{ac}}}} = L^{hb}_{i_{ac}} \times IP_a \tag{8}$$

$$MO_{a_{i_{ac}}} = MP_a \times M^b_h \tag{9}$$

where the starting offer, $SO_{a_{i_{ac}}}$, uses the "starting_price" percentual value of Figure 5, SP_a, and multiplies it by the M^b_h that represents the market price from energy bought from the main grid at hour h. The incremented offer, $IO_{a_{r_{i_{ac}}}}$, has at it bases the current highest bid of the lot at auction i, $L^{hb}_{i_{ac}}$, and

FIGURE 6. Building agents' deployment.

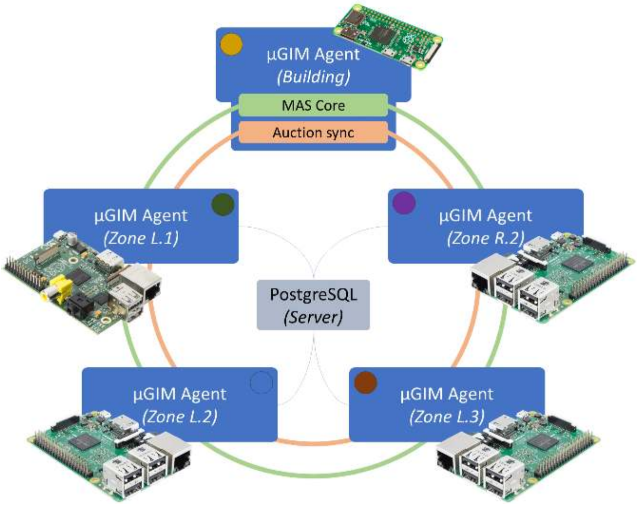


FIGURE 7. μGIM agents representing hardware and connections.

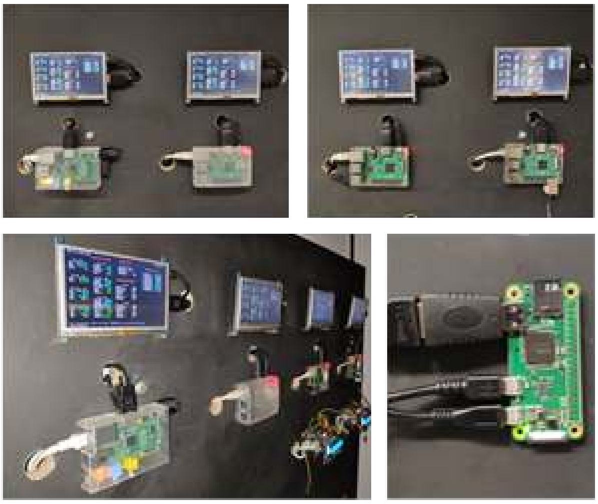


FIGURE 8. SBC installation.

the percentual value of increment, IP_a, specified in the agent configuration file. The maximum offer, $MO_{a_{iac}}$, is calculated using "max_price" value, MP_a, and the market price. Agents consider the energy prices to and from the main grid, while other prices and taxes, such as peak power cost, are currently not considered.

A second constraint is used to prevent offers in lots where the highest bid is higher than the maximum offer of agent a. If this constraint is not respected, the agent does not provide and offer.

$$L_{iac}^{hb} < MO_{a_{iac}} \tag{10}$$

If constraints (5) and (10) are respected, then agent a makes an offer (O_{a_r}) to the bid request r according to equation (11).

$$O_{a_{r_{iac}}} = \begin{cases} SO_{a_{iac}} & if\ L_{iac}^{hb} < SO_{a_{iac}} \\ MO_{a_{iac}} & if\ L_{iac}^{hb} \geq SO_{a_{iac}}\ and\ IO_{a_{r_{iac}}} > MO_{a_{iac}} \\ IO_{a_{r_{iac}}} & if\ L_{iac}^{hb} \geq SO_{a_{iac}}\ and\ IO_{a_{r_{iac}}} \leq MO_{a_{iac}} \end{cases} \tag{11}$$

According to the constraint (5), agents do not buy or bid lots bigger than the amount of energy they want to buy. This means that even with several agents willing to bid on lots, it is possible to have lots without any bid because they fall on constraints (5).

Each transaction is stored in the seller and buyer agents' databases. However, a more reliable, scalable and secure method should be applied. Distributed ledgers are a good solution that allows a decentralized approach, compliant with the proposed model. The result from peer-to-peer transactions can be validated and stored in Corda platform [34], where transactions can be added using the RPC (remote procedure call) client and/or its API.

V. BUILDING DEPLOYMENT

A total of five μGIM agents were deployed in one office building with five independent end-users. In this scenario, we have a building owner that is responsible for office rentals. Offices can be rented in sets of two or three. Each tenant and the building owner are represented by individual agents that control the building/office energy and can transact energy. Each tenant is an independent end-user with an individual energy contract with an energy provider. Although they share their physical location, they are five separate end-users that could have five different physical buildings, this would not change our case study.

Figure 6 shows the satellite image of our building, identifying the operation area of each agent. The building agent is responsible for all the common areas plus the kitchen (room 10) and rooms 11 and 16. The tenant responsible for zone L.1 has three offices. The tenant responsible for zone L.2 has three offices, one of them (room 4) is used as a server room. Zone L.3, with three rooms, is rented to another tenant. The last tenant is renting rooms 12 and 15 (i.e. zone R.2). Rooms 13 and 14 are empty and are not used or measured.

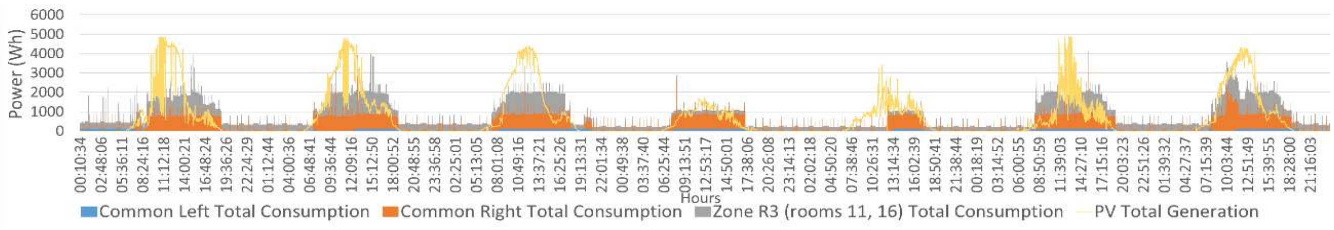


FIGURE 9. **Z.0 agent week metering.**

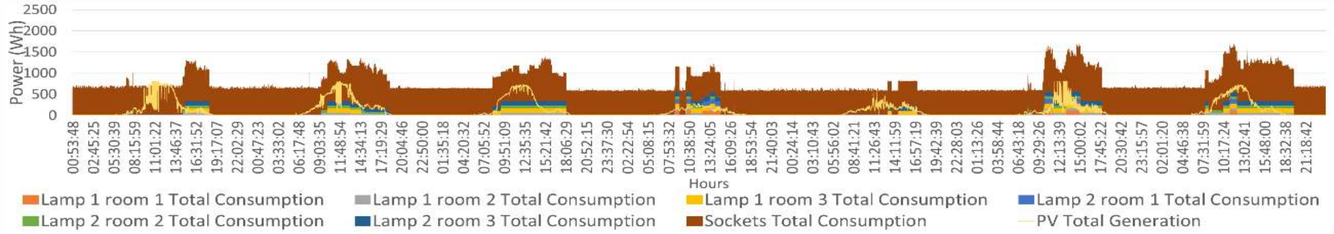


FIGURE 10. **L.1 agent week metering.**

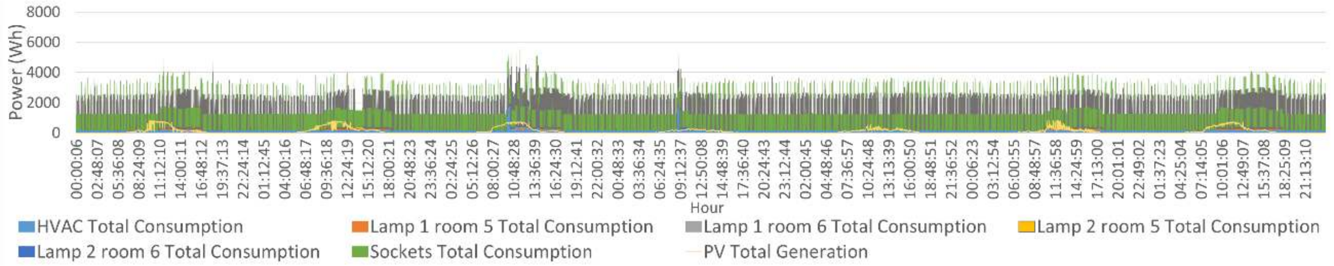


FIGURE 11. **L.2 agent week metering.**

The building has a photovoltaic peak generation of 10 kW. When a player rents a zone (i.e. a set of offices), it has access to 1 kW of generation that can be used for auto-consumption or in peer-to-peer transactions. In our case, the building's agent manages the remaining 6 kW of generation, resulting from the 10 kW less the 4 kW attributed to the tenants – 1 kW for L.1 agent, 1 kW for L.2 agent, 1 kW for L.3 agent, and 1 kW for R.2 agent.

The deployed agents are executed in five Raspberry Pi boards, shown in Figure 7. Each Raspberry Pi has one μGIM agent running in its operating system (i.e. Raspbian). Communication among agents is done using JADE framework, through TCP/IP protocol. The building's agent is executed in one Raspberry Pi Zero with a 1 GHz single-core CPU and 512 MB of RAM. The agent representing zone L.1 is executed in one Raspberry Pi with a 700 MHz single-core CPU and 512 MB. The other three agents are executed in three Raspberry Pi 3 Model B with a 1.2 GHz quad-core CPU and 1 GB of RAM.

In order to demonstrate the μGIM capabilities, Raspberry Pi Zero is configured as MAS core and the auction synchronizer. Raspberry Pi Zero manages the yellow areas of Figure 6, accommodates the JADE main container, and coordinates and manages the transactive energy auctions. This SBC is the only one running two agents: μGIM agent, and JADE directory facilitator (DF) agent.

Figure 7 presents the connections among all agents using the MAS (represented in green circular line). The beige circular line inside the green line represents the connectivity of agents regarding peer-to-peer transactions. Although the building's agent acts as the auction synchronizer, it also participants in auctions.

Agents also have a direct link to a remote PostgreSQL server. Because the storage limit of SBC is small, the stored data cannot be forever stored in the SBC. Therefore, data with more than one day is transferred to an external server where it is stored indefinitely. After the data has been stored in the external server, the μGIM agent erases the same data from the local database located inside the SBC.

In this case study, all agents are configured as presented in Figure 5. The exception is the building's agent that does not buy any energy, as the buy ‘‘baseline’’ is set to ‘‘none’’.

The monitoring and control of the energy resources, in the entire building, is done using energy analyzers and smart

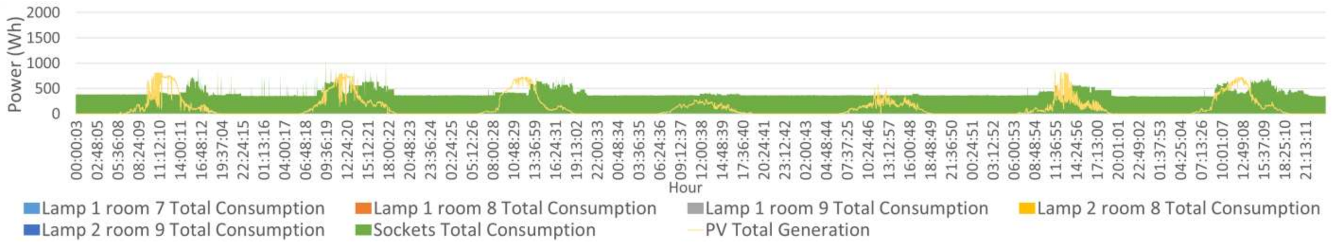


FIGURE 12. **L.3 agent week metering.**

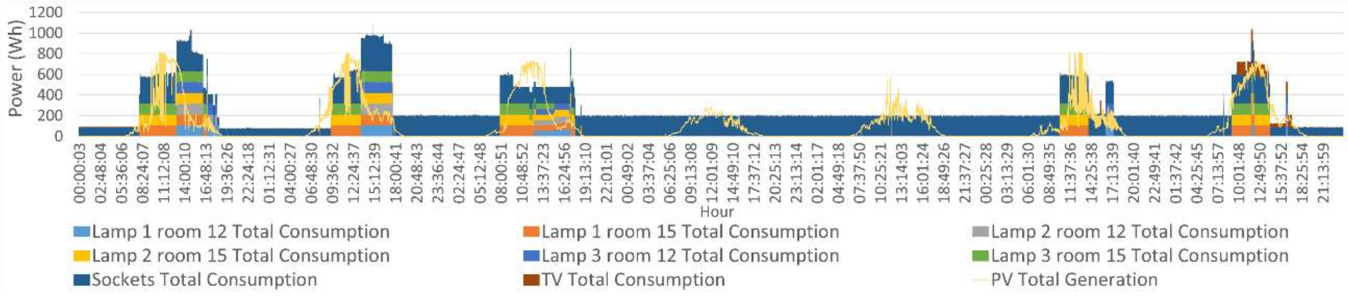


FIGURE 13. **R.2 agent week metering.**

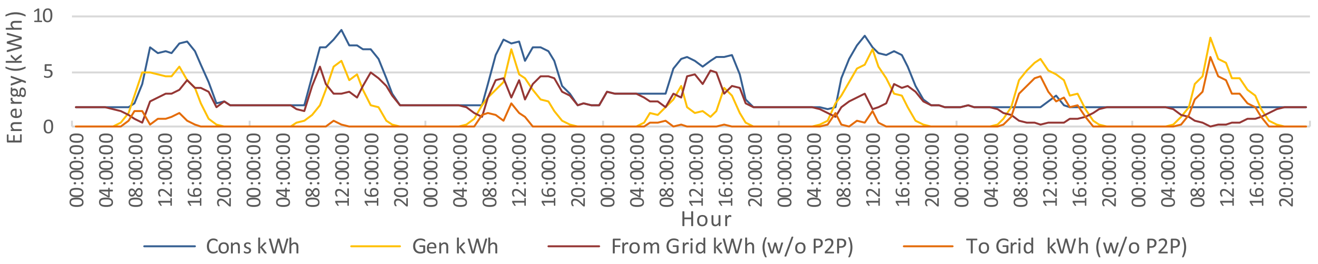


FIGURE 14. **Microgrid's weekly energy profile (without P2P transactions).**

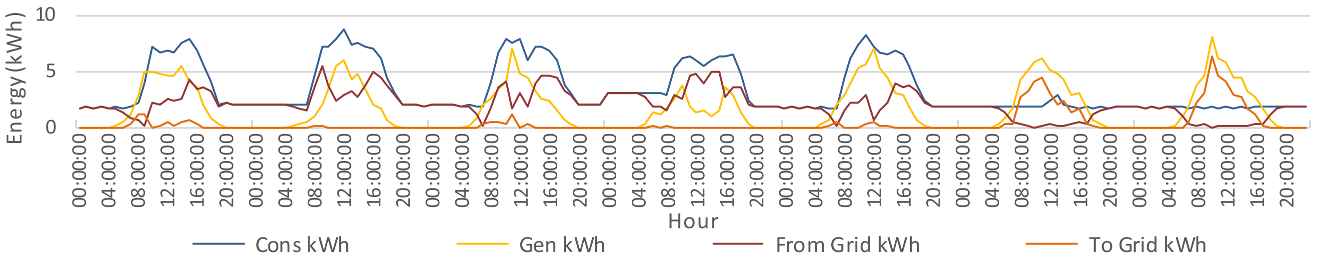


FIGURE 15. **Microgrid's weekly energy profile (with P2P transactions).**

plugs. The energy analyzers use an RS485 network with a master Programmable Logic Control (PLC). The PLC is accessed by the μGIM agents using Modbus/TCP protocol. The smart plugs, however, use TCP/IP RESTful API. Agents monitor resources each second and store their data every ten seconds. Each agent is responsible for monitoring and controlling the resources available in its area (Figure 6). In this case study, only the monitoring ability is used to measure energy consumption and generation. Because each agent manages multiple resources, each resource is monitored by an individual thread in μGIM agents.

All SBC were installed in room 14 of the building (Figure 6). In this room, it is available a board with several SBC and HDMI monitors that are used, as can be seen in Figure 8. Local monitors are not mandatory, but they were used for visualization proposes and debug. Although SBC are installed in the same physical location, they control distinct parts of the building, according to the areas presented in Figure 6.

Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13 show the weekly profile of generation and consumption in each agent. These data were collected and stored by

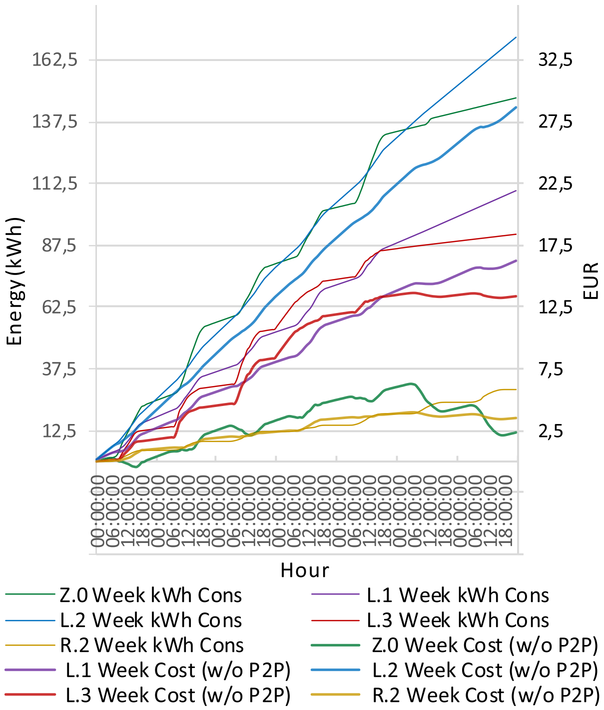


FIGURE 16. Agent's weekly consumption and energy cost (without P2P transactions).

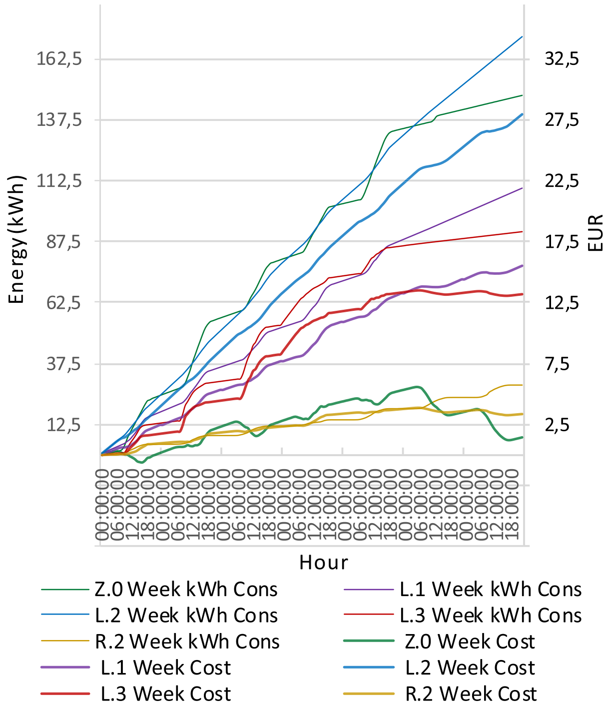


FIGURE 17. Agent's weekly consumption and energy cost (with P2P transactions).

the five μGIM agents deployed. The data regard the period between 10 April 2019 (Wednesday) and 16 April 2019 (Tuesday). The generation is represented by a yellow line and all the consumptions resources are shown as aggregated. The mentioned dataset is published under open access in [35].

Z.0 Agent represents the building's agent. It has the second higher consumption and the highest generation values (Figure 9). In zone L.1, the available generation is always lower than consumption, making this agent not capable of selling energy (Figure 10). The same scenario occurs in agent L.2, which has the highest consumption value (Figure 11). The high consumption of L.2, and its periodic consumption peaks, is the direct result of the server room located in room 4, especially its air-conditioning unit. Agents L.2 and R.2 are usually self-sustainable during the middle of the day, enabling the selling of energy to other agents (Figure 12 and Figure 13).

VI. RESULTS

This section presents the results obtained during the one week, from 3 March 2020 (Monday) to 9 March 2020 (Sunday), where the proposed TE auction model and the μGIM platform were in continuous execution without supervision. The results present data from the week with peer-to-peer (P2P) energy transaction, and without P2P transactions. The mentioned dataset is published under open access in [36]. Energy costs are related to the one week of experimentation, they are the direct result of energy bought from the grid, energy sold to the grid, energy bought in P2P auctions, and energy sold in P2P auctions.

Figure 14 shows the microgrid consumption and generation, as well as the energy bought from the main grid and energy injected/sold to the main grid. The data of Figure 14 do not consider P2P transactions. Because no P2P transactions are made, the microgrid as a whole – the combination of the five individual agents – buys and sells energy to the main grid at the same time (e.g. one agent is injecting energy will others are buying). The goal for using P2P transactions is to avoid the green line of Figure 14, indicating that energy is being sold directly to the main grid without being sold to other microgrid's agents. In Figure 15, where the results with P2P transactions are shown, the green line significantly decreases.

Figure 16 and Figure 17 show the relation, during the week, between energy consumption and energy costs in the five agents – thinner lines represent consumption while thicker lines represent costs. Especially in Z.0 agent, it is noticeable a decrease in the energy cost while maintaining the same energy consumption.

Table 1 shows the detailed results from the tested week. The table has five sections that show data related to energy, forecast errors, P2P analyses, energy costs, and P2P energy. With a higher generation, 6 kW, Z.0 agent takes the highest benefit from the P2P transactions, decreasing its energy cost by 40.8 %. But all end-users were able to decrease their energy costs. As a community, energy costs were decreased by 4.4 %. This decrease was the result of peer-to-peer transactions, the energy transacted represents 4.8 % of consumed energy in the community. In this case study, according to Figure 14, the generation was majority lower than

TABLE 1. Microgrid's overall weekly results.

		Z.0	*L.1*	*L.2*	*L.3*	*R.2*	*Microgrid*
nergy	*Consumption (kWh)*	*147.311*	*109.298*	*171.986*	*91.891*	*41.860*	*562.347*
	Generation (kWh)	*169.429*	*28.775*	*28.775*	*28.775*	*28.775*	*284.531*
Errors	*Forecast MAPE Cons.*	*10.27 %*	*10.04 %*	*9.54 %*	*9.61 %*	*9.95 %*	*9.88 %*
	Forecast MAPE Gen.	*6.26 %*	*6.57 %*	*6.52 %*	*6.50 %*	*6.75 %*	*7.16 %*
P2P analyses	*Bought (kWh)*	*0.000*	*5.427*	*13.225*	*6.237*	*2.215*	*27.104*
	Sold (kWh)	*22.705*	*0.032*	*0.000*	*1.917*	*2.450*	*27.104*
	Best choice periods	*35*	*39*	*51*	*38*	*32*	*195*
	Wrong sale periods	*4*	*1*	*0*	*1*	*3*	*9*
	Sold too much periods	*15*	*1*	*0*	*2*	*8*	*26*
	Wrong purchase periods	*0*	*0*	*0*	*0*	*1*	*1*
	Bought too much periods	*0*	*2*	*1*	*1*	*0*	*4*
	Wrong trading (kWh)	*4.320*	*0.094*	*0.003*	*0.069*	*0.250*	*4.736*
	Total number of transactions	*54*	*43*	*52*	*42*	*44*	*235*
	% of best choices	*64.81 %*	*90.70 %*	*98.08 %*	*90.48 %*	*72.73 %*	*82.98 %*
Energy costs	*Week cost*	*1.4065 €*	*15.5273 €*	*27.9279 €*	*13.1305 €*	*3.3261 €*	*61.318 €*
	Week Cost (w/o P2P)	*2.3756 €*	*16.2455 €*	*28.6661 €*	*13.3470 €*	*3.4803 €*	*64.115 €*
	Price per kWh (EUR/kWh)	*0.0095 €*	*0.1421 €*	*0.1624 €*	*0.1429 €*	*0.0795 €*	*0.1090 €*
	Price per kWh (EUR/kWh) (w/o P2P)	*0.0161 €*	*0.1486 €*	*0.1667 €*	*0.1452 €*	*0.0831 €*	*0.1140 €*
P2P energy	*Price variation (w/ and w/o P2P)*	*40.79 %*	*4.42 %*	*2.58 %*	*1.62 %*	*4.43 %*	*4.36 %*
	Trading in consumption	*0.00 %*	*4.97 %*	*7.69 %*	*6.79 %*	*5.29 %*	*4.82 %*
	Trading in generation	*13.40 %*	*0.11 %*	*0.00 %*	*6.66 %*	*8.51 %*	*9.53 %*

consumption, leaving a low margin for peer-to-peer transactions. By slightly increasing the available generation, agents could have more energy to transact among them, achieving higher decreases in energy costs.

The wrong decisions in the P2P auctions can result in a loss for the agent. There are four errors that an agent can do in auctions: a wrong sale, sold too much, a wrong purchase, and bought too much. The wrong sale and wrong purchase indicate that the forecast and the real energy consumption and generation were opposite, making the agent sell/buy energy when in fact it needed to buy/sell. The sold too much and bought too much indicate that the forecast was right, regarding the agent intention (i.e. put lots to sell, or bid in lots to buy), but the agent sold/bought too much, resulting in the need of the agent to inject the surplus energy into the main grid. From the four errors that an agent can make, the wrong purchase and the bough too much have a higher cost. The average price in P2P auctions is 0.177 EUR/kWh, and if the agent buys too much, it will need to resell it to the main grid at 0.100 EUR/kWh, representing a loss of 0.077 EUR/kWh. In the wrong sale and the sale too much errors, the loss to the agent is around 0.023 EUR/kWh, which is less than one-third of the other two errors.

The number of errors, that bring losses to agents can be decreased using better forecasting algorithms or by applying safety margins. For instance, agents should not sell 100 % of their surplus energy, but only 80 % or less. Also, learning algorithms could be applied to learn the periods and contexts when the errors occur to avoid them.

VII. CONCLUSION

The concept of transactive energy enables the active participation of all players in the smart grid. They can be part of the smart grid and take economic advantages. This paper explores the potential of transactive energy using peer-to-peer energy transactions inside an agent-based microgrid platform, considering English auctions.

The paper demonstrates how μGIM platform works and how it can be deployed in a microgrid. This agent-based management system enables the representation of each microgrid player, using low-cost, low-power and small-size single-board computers. The ability to manage the player's energy resources, while providing the communication ability of a multi-agent system, enables μGIM to be a complete microgrid management system, where all agents can cooperate and collaborate to achieve a common microgrid goal. In the presented case study, communications among agents are used to enable peer-to-peer energy transactions.

In this paper, where a peer-to-peer energy transaction model based on auctions is proposed, each μGIM agent is equipped with hour-ahead forecasting algorithms for energy consumption and generation and can participate in the peer-to-peer transaction auctions.

The μGIM platform and the peer-to-peer transaction model were deployed in an office building where five independent agents, representing offices, compose a microgrid. This work intended to measure the benefits of peer-to-peer transaction auctions inside a microgrid to decrease the demand and injection of energy from/into the main grid.

Because all agents are independent entities, they all have energy contracts with energy suppliers. However, if they transact energy among them, they can decrease energy costs. The case study, of this paper, shows that this is a possibility for the majority of the microgrid players.

The results also show that some agents have high numbers of wrong transactions, leading to bad results. This scenario could be improved by using better forecasting algorithms and by applying safety margins to transactions. Although the results of this paper, with a real case study, are very promising, they can still be improved by applying new methodologies for peer-to-peer participation.

REFERENCES

[1] (2015). *The GridWise Architecture Council, GridWise Transactive Energy Framework Version 1.0*. Accessed: Mar. 20, 2018. [Online]. Available: https://www.gridwiseac.org/pdfs/te_framework_report_pnnl-22946.pdf

[2] K. Kok and S. Widergren, "A society of devices: Integrating intelligent distributed resources with transactive energy," *IEEE Power Energy Mag.*, vol. 14, no. 3, pp. 34–45, May/Jun. 2016, doi: 10.1109/MPE.2016.2524962.

[3] *Digital America*, Consum. Technol. Assoc., Arlington, TX, USA, 2015, p. 85.

[4] *Smart Home Report 2018—Control and Connectivity, Smart Home Report 2018—Control and Connectivity*, Statista, Hamburg, Germany, 2018.

[5] Y. Han, K. Zhang, H. Li, E. A. A. Coelho, and J. M. Guerrero, "MAS-based distributed coordinated control and optimization in microgrid and microgrid clusters: A comprehensive overview," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 6488–6508, Aug. 2018, doi: 10.1109/TPEL.2017.2761438.

[6] Z. Li, C. Zang, P. Zeng, H. Yu, and H. Li, "MAS based distributed automatic generation control for cyber-physical microgrid system," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 1, pp. 78–89, Jan. 2016, doi: 10.1109/JAS.2016.7373765.

[7] F. Chen, M. Chen, Q. Li, K. Meng, J. M. Guerrero, and D. Abbott, "Multiagent-based reactive power sharing and control model for islanded microgrids," *IEEE Trans. Sustain. Energy*, vol. 7, no. 3, pp. 1232–1244, Jul. 2016, doi: 10.1109/TSTE.2016.2539213.

[8] T. W. Overton. (Jan. 2014). POWER Magazine. New York University Cogeneration Plant, New York, NY, USA. Accessed: Aug. 7, 2017. [Online]. Available: http://www.powermag.com/new-york-university-cogeneration-plant-new-york-city

[9] B. Washom, J. Dilliot, D. Weil, J. Kleissl, N. Balac, W. Torre, and C. Richter, "Ivory tower of power: Microgrid implementation at the University of California, San Diego," *IEEE Power Energy Mag.*, vol. 11, no. 4, pp. 28–32, Jul./Aug. 2013.

[10] A. Villanueva, V. Fung, J. Worrall, and J. Trueblood. (Dec. 2014). *Camp Pendleton Fractal Grid Demonstration, Energy Research and Development Division Final Project Report*. Accessed: Mar. 29, 2019. [Online]. Available: https://www.energy.ca.gov/2016publications/CEC-500-2016-013/CEC-500-2016-013.pdf

[11] Naval Facilities Engineering Command. (Dec. 31, 2015). *Smart Power Infrastructure Demonstration for Energy Reliability and Security Technology Transition Final Public Report*. Accessed: Mar. 29, 2019. [Online]. Available: https://www.energy.gov/sites/prod/files/2016/03/f30/spiders_final_report.pdf

[12] LO3ENERGY. (Apr. 24, 2018). *EXERGY—Business Whitepaper*. Accessed: Mar. 29, 2019. [Online]. Available: https://lo3energy.com/wp-content/uploads/2018/04/Exergy-BIZWhitepaper-v11.pdf

[13] D. Cardwell. (Mar. 13, 2017). *Solar Experiment Lets Neighbors Trade Energy Among Themselves*. The New York Time. Accessed: Mar. 23, 2019. [Online]. Available: https://www.nytimes.com/2017/03/13/business/energy-environment/brooklyn-solar-grid-energy-trading.html

[14] B. Akyol, C. H. Allwardt, Z. W. Beech, J. B. Chapman, J. N. Haack, S. Katipamula, R. G. Lutes, and K. E. Monson. (Jun. 2016). *VOLTTRONtm 2016*. Accessed: Mar. 23, 2019. [Online]. Available: https://volttron.org/sites/default/files/publications/PNNL-25499_VOLTTRON_2016.pdf

[15] U. Herberg, D. Mashima, J. G. Jetcheva, and S. Mirzazad-Barijough, "OpenADR 2.0 deployment architectures: Options and implications," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Venice, Italy, Nov. 2014, pp. 782–787, doi: 10.1109/SmartGridComm.2014.7007743.

[16] C. D. Corbin, A. Makhmalbaf, S. Huang, V. V. Mendon, M. Zhao, S. Somasundaram, G. Liu, H. Ngo, and S. Katipamula. (Dec. 2016). *Transactive Control of Commercial Building HVAC Systems*. Accessed: Mar. 23, 2019. [Online]. Available: https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-26083.pdf

[17] S. Eisele, I. Mardari, A. Dubey, and G. Karsai, "RIAPS: Resilient information architecture platform for decentralized smart systems," in *Proc. IEEE 20th Int. Symp. Real-Time Distrib. Comput. (ISORC)*, Toronto, ON, Canada, May 2017, pp. 125–132, doi: 10.1109/ISORC.2017.22.

[18] S. Eisele, A. Dubey, G. Karsai, and S. Lukic, "WiP abstract: Transactive energy demo with RIAPS platform," in *Proc. ACM/IEEE 8th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Pittsburgh, PA, USA, Apr. 2017, pp. 91–92.

[19] Y. Du, H. Tu, S. Lukic, A. Dubey, and G. Karsai, "Distributed microgrid synchronization strategy using a novel information architecture platform," in *Proc. IEEE Energy Convers. Congr. Exposit. (ECCE)*, Portland, OR, USA, Sep. 2018, pp. 2060–2066, doi: 10.1109/ECCE.2018.8557695.

[20] M. H. Cintuglu, T. Youssef, and O. A. Mohammed, "Development and application of a real-time testbed for multiagent system interoperability: A case study on hierarchical microgrid control," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1759–1768, May 2018, doi: 10.1109/TSG.2016.2599265.

[21] Z. Liu, Q. Wu, S. Huang, and H. Zhao, "Transactive energy: A review of state of the art and implementation," in *Proc. IEEE Manchester PowerTech*, Jun. 2017, pp. 1–6, doi: 10.1109/PTC.2017.7980892.

[22] H. Hao, C. D. Corbin, K. Kalsi, and R. G. Pratt, "Transactive control of commercial buildings for demand response," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 774–783, Jan. 2017, doi: 10.1109/TPWRS.2016.2559485.

[23] C. Zhang, J. Wu, C. Long, and M. Cheng, "Review of existing peer-to-peer energy trading projects," *Energy Procedia*, vol. 105, pp. 2563–2568, May 2017, doi: 10.1016/j.egypro.2017.03.737.

[24] B. Brandherm, J. Baus, and J. Frey, "Peer energy cloud-civil marketplace for trading renewable energies," in *Proc. 8th Int. Conf. Intell. Environ.*, Guanajuato, Mexico, 2012, pp. 375–378, doi: 10.1109/IE.2012.46.

[25] M. Fry. (Nov. 20, 2018). *CASE STUDY: Learn More About Our Live Project With BCPG in Bangkok, Thailand*. Accessed: Mar. 29, 2019. [Online]. Available: https://medium.com/power-ledger/case-study-learn-more-about-our-live-project-with-bcpg-in-bangkok-thailand-ab7a31c8b464

[26] A. Singh, A. T. Strating, N. A. R. Herrera, D. Mahato, D. V. Keyson, and H. W. van Dijk, "Exploring peer-to-peer returns in off-grid renewable energy systems in rural India: An anthropological perspective on local energy sharing and trading," *Energy Res. Social Sci.*, vol. 46, pp. 194–213, Dec. 2018, doi: 10.1016/j.erss.2018.07.021.

[27] C. Zhang, J. Wu, Y. Zhou, M. Cheng, and C. Long, "Peer-to-peer energy trading in a microgrid," *Appl. Energy*, vol. 220, pp. 1–12, Jun. 2018, doi: 10.1016/j.apenergy.2018.03.010.

[28] M. R. Alam, M. St-Hilaire, and T. Kunz, "Peer-to-peer energy trading among smart homes," *Appl. Energy*, vol. 238, pp. 1434–1443, Mar. 2019, doi: 10.1016/j.apenergy.2019.01.091.

[29] T. Morstyn, A. Teytelboym, and M. D. Mcculloch, "Bilateral contract networks for peer-to-peer energy trading," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2026–2035, Mar. 2019, doi: 10.1109/TSG.2017.2786668.

[30] F. H. Malik and M. Lehtonen, "A review: Agents in smart grids," *Electr. Power Syst. Res.*, vol. 131, pp. 71–79, Feb. 2016, doi: 10.1016/j.epsr.2015.10.004.

[31] M. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. Hoboken, NJ, USA: Wiley, 2009.

[32] L. Gomes, J. Spínola, Z. Vale, and M. J. Corchado, "Agent-based architecture for demand side management using real-time resources' priorities and a deterministic optimization algorithm," *J. Cleaner Prod.*, vol. 241, 2019, doi: 10.1016/j.jclepro.2019.118154.

[33] L. Gomes, Z. Vale, and J. M. Corchado, "Microgrid management system based on a multi-agent approach: An office building pilot," *Measurement*, vol. 154, Mar. 2020, Art. no. 107427, doi: 10.1016/j.measurement.2019.107427.

[34] *Corda Documentation*. Accessed: Mar. 23, 2019. [Online]. Available: https://docs.corda.net/index.html

[35] L. Gomes, "uGIM: Week monitorization data of a microgrid with five agents (10/04/19-16/04/19), version 1.0," Zenodo, 2019, doi: 10.5281/zenodo.2868129.

[36] L. Gomes, "uGIM: A week with peer-to-peer transactions (02/03/2020-08/03/2020), Version 0.1.0," Zenodo, 2019, doi: 10.5281/zenodo.3707578.

LUIS GOMES received the B.Sc. and master's degrees in computer engineering from the Polytechnic of Porto (ISEP/IPP), Porto, Portugal, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree with the GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto. His research interests include multiagent systems, microgrids, and demand side management.

ZITA A. VALE (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Porto, Porto, Portugal, in 1993. She is currently a Professor with the Polytechnic Institute of Porto, Portugal. Her research interests focus on artificial intelligence applications, smart grids, electricity markets, demand response, electric vehicles, and renewable energy sources.

JUAN MANUEL CORCHADO (Member, IEEE) was born in Salamanca, Spain, in 1971. He received the Ph.D. degree in computer sciences from the University of Salamanca, and the Ph.D. degree in artificial intelligence from the University of the West of Scotland. He was the Vice President for Research and Technology Transfer, from 2013 to 2017, and the Director of the Science Park with the University of Salamanca, where he was also the Director of the Doctoral School, until 2017. He has been elected twice as the Dean of the Faculty of Science with the University of Salamanca. He has been a Visiting Professor with the Osaka Institute of Technology, since 2015, and a Visiting Professor with University Teknologi Malaysia, since 2017. He is currently a Full Professor with the Chair, University of Salamanca. He is also a member of the Advisory Group on Online Terrorist Propaganda of the European Counter Terrorism Centre (EUROPOL). He is also the Director of the Bioinformatics, Intelligent Systems and Educational Technology (BISITE) Research Group, which he created, in 2000. He is also the President of the IEEE Systems, Man and Cybernetics Spanish Chapter and the Academic Director of the Institute of Digital Art and Animation, University of Salamanca. He also oversees the master's programs in digital animation, security, mobile technology, community management and management for TIC Enterprises with the University of Salamanca. He is also an Editor and the Editor-in-Chief of specialized journals, such as the *Advances in Distributed Computing and Artificial Intelligence Journal*, the *International Journal of Digital Contents and Applications*, and the *Oriental Journal of Computer Science and Technology*.

• • •