An Open MAS/IoT-Based Architecture for Large-Scale V2G/G2V

Charilaos Akasiadis[1], Georgios Iatrakis[2], Nikolaos Spanoudakis[3](✉), and Georgios Chalkiadakis[2]

[1] Institute of Informatics and Telecommunications, NCSR 'Demokritos', Athens, Greece
cakasiadis@iit.demokritos.gr

[2] School of Electrical and Computer Engineering, Technical University of Crete, Kounoupidiana, Greece
{giatrakis,gchalkiadakis}@isc.tuc.gr

[3] School of Production Engineering and Management, Technical University of Crete, Kounoupidiana, Greece
nispanoudakis@isc.tuc.gr

Abstract. In this paper we put forward an open multi-agent systems (MAS) architecture for the important and challenging to engineer vehicle-to-grid (V2G) and grid-to-vehicle (G2V) energy transfer problem domains. To promote scalability, our solution is provided in the form of modular microservices that are interconnected using a multi-protocol Internet of Things (IoT) platform. On the one hand, the low-level modularity of Smart Grid services allows the seamless integration of different agent strategies, pricing mechanisms and algorithms; and on the other, the IoT-based implementation offers both direct applicability in real-world settings, as well as advanced analytics capabilities by enabling digital twins models for Smart Grid ecosystems. We describe our MAS/IoT-based architecture and present results from simulations that incorporate large numbers of heterogeneous Smart Grid agents, which might follow different strategies for their decision making tasks. Our framework enables the testing of various schemes in simulation mode, and can also be used as the basis for the implementation of real-world prototypes for the delivery of large-scale V2G/G2V services.

Keywords: Internet of things · Open multi-agent systems · Smart grid

1 Introduction

In the emerging Smart Grid [2], energy and information flow towards all possible directions over distribution and transmission networks. As such, buildings but also vehicles become active consumers and producers of energy, and need to be integrated into the Grid. Not only is the Smart Grid an electricity network with diverse consumers and producers, it is also a dynamic marketplace where heterogeneous devices appear and need to connect [9]. To date, several Smart

F. Dignum et al. (Eds.): PAAMS 2022, LNAI 13616, pp. 3–14, 2022.
https://doi.org/10.1007/978-3-031-18192-4_1

Grid-related business models and information systems' architectures have been proposed, but they do not always adhere to particular standards [4]. This is normal, as the energy markets involved can be global, regional, or isolated; can be based mostly on renewable energy or not; and can be regulated by a public authority or allow dynamic pricing based on demand and offer.

Such energy markets naturally reflect systems where not one player can force others to use her products; players or stakeholders can come along their own business models; and stakeholders can have diverse goals in negotiating their consumption and offer. Moreover, these systems allow for pro-activeness of the players who pursue their goals and sociability—as they can form dynamic partnerships or coalitions, but also react and/or adapt to a changing dynamic environment [13]. In addition, it is natural for participants to be generally able to freely join and leave the system at any time. All these characteristics point to agent technology and open *multiagent systems (MAS)* in particular [20].

At the same time, the advances in the domain of the Internet of Things (IoT) allow the deployment of such approaches in the real world, as IoT offers a networking layer that interconnects distributed resources, e.g. power meters and other sensors, charging controllers and similar actuators, decision support agents and various processing services [5]. A key IoT concept is that these resources, although heterogeneous, are interoperable in the sense that they exchange information and reconfigure particular parameters, crucial for their operation.

To the best of our knowledge, however, existing approaches for the Vehicle-to-Grid (V2G)/ Grid-to-Vehicle (G2V) problem do not provide functional open prototypes offering such features, or adequately exploit existing engineering MAS research paradigms. In an open system, diverse agents representing stakeholders need to use predefined protocols to interact; but also need to work the protocols with their own algorithms and/or goals. Given this, the main contribution of this paper is a novel MAS/IoT architecture we put forward for the V2G/G2V domain. Our architecture allows the different stakeholders to reuse existing agents in new deployments, or to develop new ones, according to respective goals. We propose the instantiation of such a system using SYNAISTHISI, a research-oriented IoT platform deployed in docker containers, which allows agents to connect and communicate using the Message Queuing Telemetry Transport (MQTT) publish/subscribe protocol [1]. The validity of the approach is illustrated via simulation experiments with two different dynamic pricing mechanisms and three charging scheduling algorithms inspired by the existing literature.

In the rest of this paper, we first present the necessary background and discuss related work (Sect. 2). Then, Sect. 3 presents our V2G/G2V-specific MAS-based architecture and the roles that the different agents have. Section 4 details the system development process, along with the IoT communications infrastructure and agent interaction protocols. Following that, in Sect. 5, we evaluate the applicability of our approach with realistic use case scenarios of interest. Finally, Sect. 6 concludes this paper.

2 Background and Related Work

Recent trends indicate that, in the near future, large numbers of EVs will penetrate into the electricity markets resulting to different demand patterns, altered enough to disrupt the stability and reliability of existing power networks [6]. To overcome this, researchers have introduced "smart charging", or Grid-to-Vehicle (G2V) approaches, where charging might not be initiated instantly upon EV connection, but get delayed due to various factors [3], e.g., renewable production levels, demand from other EVs, pricing, etc. Complementary to G2V, the Vehicle-to-Grid (V2G) approach takes advantage of the electricity storage capabilities of EV batteries, and allows their controlled discharging for supporting the Grid during times of energy supply shortage [15].

Research has focused on combining simulators with (possibly smaller-scale) real-world trials for the delivery of V2G and G2V services. For example, XBOS-V [12] is a system for controlling plug-in EV charging in residential and small commercial sites. RISE-V2G is an implementation of the V2G communication interface ISO 15118, i.e. a standardized communication method, which provides lower level connection infrastructure between electric vehicles and charging stations. Similar examples are the Open Charge Point Protocol (OCPP), the Open Charge Point Interface (OCPI), and the Open Smart Charging Protocol (OSCP). OpenV2G [7] implements the necessary components of the V2G public key infrastructure. The focus of the approach is to securely connect electric vehicles and charging stations and provide simulation capabilities. GEM [17], another approach, simulates the operation on both the mobility and the electricity domains. However, the simulation approach followed represents a higher level and does not include particular stakeholder types, such as a station recommender. ACN-Sim [10] is a tool for managing battery charging and performs a user- rather than a grid-level analysis.

SYNAISTHISI IoT is a research-oriented platform that brings together open-source frameworks into a unified solution with many desirable properties [1]. In particular, services and sub-modules come in docker containers, allowing scalable and operating system independent deployments. For user developed services, the platform can act as a message-oriented middleware enabling their communication and orchestration. To that end, multiple protocols are supported and can be translated to one another instantly, so that agent and service heterogeneity with respect to their implementation details is sufficiently captured. Moreover, authentication and authorization is supported for restricting access to topics holding private information, and semantic descriptions of services and exchanged information allow more sophisticated processing and knowledge extraction. These features fulfill the requirements of our proposed open system, and also enable the reusability of services for fast system redeployment at different locations.

3 System Architecture

In this section we provide an overview of our architecture. We assume that agents coexist in a microgrid infrastructure that can be interconnected with

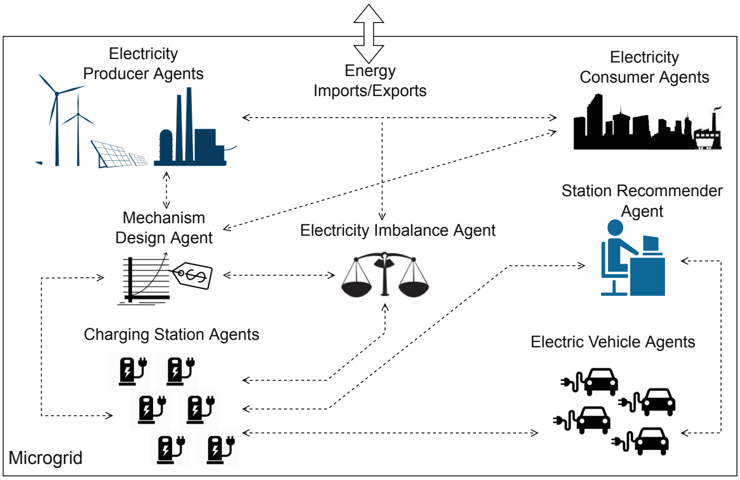


Fig. 1. Overview of the proposed MAS architecture for the V2G/G2V problem.

other parts of the Smart Grid through distribution and transmission networks. When a microgrid requires power that can not be generated locally, it can import it, while, when it has a local energy surplus, it can export it to the Grid and create additional profits for its producers, according to energy market regulations [9]. Figure 1 provides an overview of the agents and their interactions.

In particular, the agent types in our system are: the *(a)* Electric Vehicle agents (EV), the *(b)* Charging Station agents (CS), the *(c)* Electricity Producer agents (EP), and the *(d)* Electricity Consumer agents (EC). We also assume the existence of a regulatory service (possibly a for-profit private service), that consists of the following agents: *(i)* a Station Recommender (SR), *(ii)* an Electricity Imbalance (EI), and *(iii)* a Mechanism Design (MD). In what follows, we refer to this service by its three distinct agents separately. Note that each agent type may consist of certain "private" sub-modules, whose specific functionality can further differentiate agent behaviors.

EV agents aim to optimize a utility function set by the EV owner— e.g., always have enough energy to realize the next trip, achieve so by the minimum cost, etc. The *EV agent* monitors the driver's activities, models and predicts her future behavior and needs, and can contact a charging station to schedule battery charging, seek profit from participation in V2G activities, and engage in negotiations with the charging stations. Building blocks can be preference elicitation modules, responsible for monitoring the habits and behavior of the driver, and perhaps even forecast future preferences; user interfaces accessible by humans either via mobile devices or the vehicle's dashboard, to operate respective procedures and monitor their conduct, for example payments, negotiations, or browse and select recommendations. Such agents can implement alternative strategies for automatically selecting a charging schedule according to predefined needs of each driver, e.g. less cost, quicker availability, charging network preferences, location-based selection etc. An *EV agent* communicates with the *SR agent* to receive recommendations and the *CS agents* to reserve station slots.

Next, *CS agents* manage the physical gateways (i.e. connectors, parking slots) by which EVs connect to the grid and create profit by charging their batteries. They can also negotiate with *EV agents* regarding an existing charging agreement, to change some of the parameters so as be able to schedule the charging of additional vehicles. This leads to better utilization of the station infrastructure, and maximizes its profit. A *CS agent* may contain a charging scheduling module, the algorithmic component responsible for schedule charging/discharging activities over a predefined planning horizon; a negotiation decision making module for conducting negotiations; a pricing module that calculates costs and payments; and a preference elicitation module that monitors charging slots usage and updates the prices for each of them according to the needs of the station owner; A *CS Agent* communicates with the *SR*, the *MD*, the *EI*, and *EV* agents.

The *SR agent* recommends to EVs a subset of the available CS and charging slots that match most with their preferences (e.g. duration, distance). This agent can be also augmented to take into account various grid constraints in order to, e.g., help avoid herding effects. It consists of a recommendations engine module, an EV repository module that stores information about the past EV behavior in order to utilize it for future recommendations, and a charging station repository of registered CSs. It communicates with the *CS* and *EV* agents.

The *EI agent* aggregates data from the *EP*, *CS*, and *EC* agents regarding their expected energy profiles, and calculates the periods of electricity shortage and surplus. Then, it provides the imbalance levels to all interested parties, for them to plan their consumption and production activities. It employs a constraints extraction module that incorporates various measures and methods that could be relevant in such a scenario, and calculates electricity imbalance over a predefined planning horizon; stations, producers, and consumers repositories.

The *MD agent* represents an intermediate trusted third party entity, responsible for calculating dynamic prices and managing the payments of the various contributor types. Its goal is to assign appropriate, and possibly personalized rates for energy consumption and production by *CS*, *EC*, and *EP* agents. It can be equipped with pricing mechanisms that incentivize agents to be truthful regarding their statements for expected values, as well as their actual behavior.

Finally, the various *EP* and *EC* agents predict and periodically report expected production and consumption levels respectively, and their confidence on such predictions. These types of agents typically communicate with *EI* and *MD* agents. Every agent type may also have user interfaces, either for mere monitoring in case of fully autonomous operation, or for additional human interaction in semi-automatic or manual modes.

4 Agent Interactions

We followed a methodological approach to system analysis and design, based on the Agent Systems Engineering Methodology (ASEME), which has been employed in the past for modeling Ambient Intelligence applications [18] and

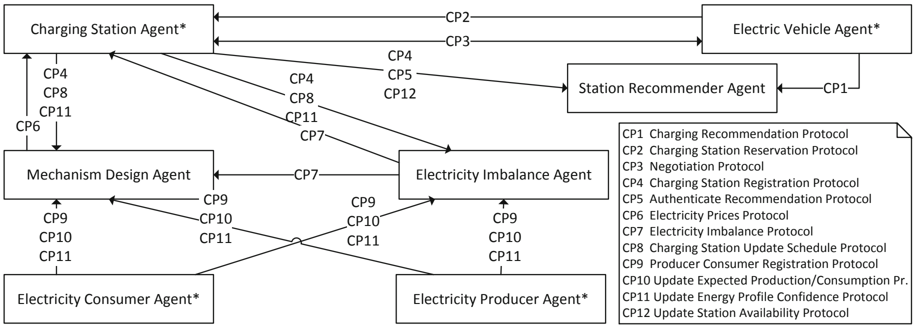


Fig. 2. The proposed architecture. (*) denotes agent types with multiple instances. Arrows start from the agent that initiates the protocol and point to the receiver agents.

referred to by the literature on modeling IoT-based MAS [16,21]. ASEME builds on existing languages, such as Unified Modeling Language (UML) and state charts, in order to represent system analysis and design models. It is agent architecture- and agent mental model- independent, allowing the designer to select the architecture type and the mental attributes of the agent, thus supporting heterogeneous agent architectures. Moreover, ASEME puts forward a modular agent design approach and uses the so-called intra-agent and inter-agent control concepts. The first defines the agent's behavior by coordinating the different modules that implement its capabilities, while the latter defines the protocols that govern the coordination of the society of the agents.[1]

In this sense the cooperation protocols were modeled as state charts. Figure 2 shows the agent types along with the protocols used for their cooperation.

CP1 Charging Recommendation: Initiated whenever an EV needs to schedule a charging session. The EV submits its preference and location to the SR and receives a list of recommended CSs and slots.

CP2 Charging Station Reservation: Follows right after CP1, for the EV to reserve the selected charging slot.

CP3 Negotiation: Optional, initiated after CP2, whenever the CS or the EV need to reschedule a charging slot.

CP4 Charging Station Registration: Registration of a new CS into the system and informs the MD, the EI and the SR about its specifications.

CP5 Authenticate Recommendation: Follows right after CP2. The EV sends the recommendation it selected to the CS, and CS requests from to the SR its validation.

CP6 Electricity Prices: Follows after CP7. Used by the MD to update prices and broadcast these to every CS.

CP7 Electricity Imbalance: Follows right after CP10 or CP8, if the expected production or consumption changes, the EI broadcasts the updated values to the MD and every CS.

[1] More detailed descriptions of the inter- and intra-agent control and a detailed description of the protocols, including the message syntax and semantics, can be found in our online repository: https://github.com/iatrakis/IoT-V2G-G2V.

CP8 Charging Station Update Schedule: Follows after CP5. The CS creates an updated *energy schedule* after an EV recommendation is authenticated, and communicates it to the EI and the MD.

CP9 Producer Consumer Registration: Registers new producers and consumers. The new stakeholder informs EI and MD about its type.

CP10 Update Expected Production/Consumption: Triggers at the beginning of each day. All producers and consumers inform the EI and MD agents about the next day's expected production and consumption.

CP11 Update Energy Profile Confidence: Triggers at the start of each day. All producers and consumers inform the EI and MD agents about their confidence regarding their forecasts (CP10).

CP12 Update Station Availability: Follows right after CP2. The CS updates its charging slot availability after each new reservation, and informs the SR.

Now, each agent is implemented in a different program that is deployed in an independent docker container, either hosted in cloud infrastructure, or locally in each stakeholder's premises. Moreover, to support research, we can set up simulations to test and evaluate different agent strategies and algorithms. This can be achieved by implementing additional orchestrator scripts that take advantage of the IoT platform's API for registering, deploying, and configuring services in batches, as well as for logging the actions and outcomes of each agent. We also need to define the duration of a simulation hour in actual time—e.g., two seconds correspond to one simulated hour, to configure the agent implementations accordingly. Similarly, each agent provides data regarding demand/charging preferences. In an actual system deployment though, the required data would be obtained in real-time, via sensor measurements, or user input forms.

Our implementation is based on the SYNAISTHISI platform, however any other IoT platform solution offering similar features could be used as well. We chose this particular one for a number of reasons. From a user perspective, it has a non-commercial license and can be used for research purposes, and allows developers to create new services of their own and integrate them into more complex applications; and from a technical perspective, it supports many application layer protocols (MQTT,[2] HTTP/REST,[3] etc.), it can be easily deployed as docker containers offering this way interoperability with other software and scalability for large-scale deployments. Also, it employs user authentication and authorization processes to restrict open access for private information.

The service interconnection is realized with the exchange of messages, in our case using the MQTT publish/subscribe protocol. Each service can subscribe to topics in order to receive messages, or publish to other topics where other services have subscribed to, for sending information and commands. To receive or send data, from and to particular topics, the service owner must possess appropriate access rights, which can be managed via the platform's GUI. The same holds for the deployment and the execution monitoring of the deployed

[2] MQTT is an OASIS standard messaging protocol for the Internet of Things, mqtt.org.

[3] REpresentational State Transfer (REST) over Hypertext Transfer Protocol (HTTP).

services. In case of mobile assets such as EVs, a wireless internet connection is required in order for the messages to be exchanged. For charging stations and the various Supervisory Control and Data Acquisition (SCADA) systems, appropriate connectors can offer an interface for the platform interconnection, provided that these too are connected via the internet.

4.1 Implemented Agent Strategies

For the purposes of evaluation via simulation, we need to test different methods and compare their effects on the system in simulation mode. To this end, we implemented two pricing algorithms used by the MD agent to observe how they contribute to grid stability, i.e. to reducing the energy surplus and deficit peaks. We also implemented three scheduling approaches that determine when and how much energy is exchanged between CS and EV agents.

Price Calculation Algorithms for the Mechanism Design Agent:

A) *NRG-Coin pricing algorithm:* This mechanism is inspired by the one in [14], and aims at incentivizing stakeholders to balance supply and demand.
B) *Adaptive pricing algorithm:* According to this mechanism proposed in [19], we estimate the evaluation of energy with respect to the cost induced by the EV agents. The mechanism can adjust prices to motivate agents to charge their EVs when there is an energy surplus on the grid.

Charging Scheduling Algorithms:

A) *First slot:* In this case, EVs charge their battery in the first available time interval, without taking into account if prices are better or worse.
B) *Lowest Prices:* In this approach, EVs are trying to reduce charging costs by choosing time periods that the energy prices are the lowest possible.
C) *V2G:* In this case EVs are able to discharge their batteries when the prices are high to provide load to the rest of the grid, and then charge it back when prices are lower, nevertheless within the periods that EVs are connected to a charger. For this purpose and inspired by [8] and [11], we used linear programming to minimize an objective function representing charging costs in the presence of constraints regarding the EV preferences and charging specifications.

5 Experimental Evaluation

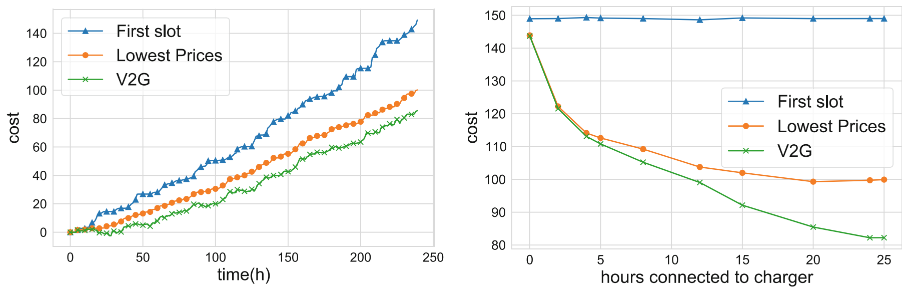
In this section we show four use cases that illustrate the applicability of our proposed architecture. The use cases provide comparative evaluations of the implemented strategies discussed in the previous section. All agent implementations are in Python, while the datasets that we used are based on a collection of real data from a number of publicly available online resources;[4] and the duration of each simulation is 10 days. The simulations were performed on a PC with an AMD Ryzen 5 1500X @ 3.5 GHz processor and 8 GB of RAM.

[4] Specifically, consumption and production data originate from the ENTSOE platform, and EV data from the MyElectricAvenue project.

5.1 Simulating Algorithms and Mechanisms

The first use case is employed to compare the different EV charging scheduling methods, using the *NRG-Coin* pricing mechanism. We remind the reader that these methods are charging *(i)* during the first slots that the EV gets connected to a charger, *(ii)* during intervals with the lowest price for consumption, and *(iii)* with V2G capability, where the EV can also sell back to the grid some of the stored energy and recharge later, provided that the price difference between the discharge and recharge intervals generates profit. Figure 3a shows the average cumulative EV costs for the entire planning horizon. As we can see, the highest cost for the EV is given by the *first slot* method, which is expected as in this case the EV agent chooses to charge immediately without considering the energy price. By adopting the *Lowest Prices* method, the total cost for EV charging drops about 33% by the end of the time horizon. Finally, by allowing *V2G* operations, the charging costs drop even more, 15% lower than those of *Lowest Prices*, and by 43% compared to the *first slot* method.

Next, we account for the impact of the different charging scheduling methods on the aggregate energy imbalance. As a baseline, we consider grid imbalance without the EV demand. We calculate the sum of the absolute imbalance values among the intervals, the sum of only the positive imbalance intervals (i.e., the total exported or "wasted" energy), and the sum of only the negative intervals (i.e., the total energy imports). Table 1 shows the significant impact of EVs strategy on the energy imbalance. When using the *first slot* method, EVs affect the system negatively, by increasing the total imbalance and adding more than double to the energy that has to be produced to meet demand in the grid. In parallel, the amount of energy wasted drops, since EVs consume energy that otherwise would not be consumed. In the case of *Lowest Prices* method, the imbalance tangibly drops, and the available energy that is utilized and does not get wasted, increases by half (specifically, by 45.6%). The imports are also increased, due to the additional demand of the 100 EVs and their occasional need to charge their batteries to continue their trip without caring about high energy



(a) Average Cumulative Cost per EV for different charging scheduling methods.

(b) Cost comparison of varying time periods that EVs are connected to chargers.

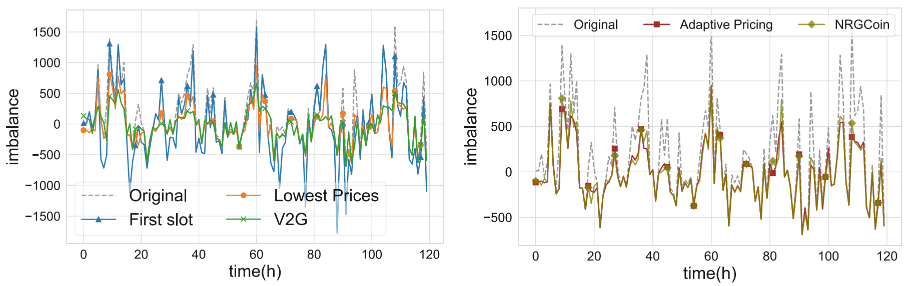
Fig. 3. Charging cost variation in different scenarios.

Table 1. Energy differences on charging scheduling methods compared to the "no EVs" baseline. The MAPE of the original imbalance curve is 63.9%.

Method	Imbalance	Wasted	Imported	MAPE
First slot	+7.0%	−21.8%	+104.2%	−12.4%
Lowest prices	−31.4%	−45.6%	+16.4%	−44.5%
V2G	−37.3%	−49.1%	+2.5%	−55.7%

prices and energy shortage of the grid. An even better picture is obtained when *V2G* comes into play, with even lower imbalance (higher imbalance reduction, reaching 37.3%); less energy wasted (waste reduced by 49.1%; while imports are increased by only a very small rate (specifically, by 2.5%). Moreover, it achieves a larger reduction in the *Mean Absolute Percentage Error (MAPE)*, than the other two methods. MAPE measures the difference of the induced imbalance from a totally flat curve with a value of zero, which resembles perfect matching between supply and demand. This is clearly visible when plotting the imbalance across the time horizon for each method, as we do in Fig. 4a. Indeed, it is noteworthy that *V2G* induces smaller peaks in the imbalance between demand and supply than the rest of the methods.

In the second use case, we measure the total cumulative cost of EVs, when increasing the duration of connection to chargers by 24 h compared to the original data, by following the three different charging scheduling methods of the first use case. The results of Fig. 3b show that by increasing the duration of connection, the *Lowest Price* and *V2G* methods manage to gradually reduce the battery charging costs. This happens since the longer an EV is connected to a charger, the higher probability it has to find the most advantageous intervals to buy energy at from the grid— and also to sell it back to the grid in the case of V2G. As anticipated, again, the *V2G* method leads to lower charging costs than the



(a) Imbalance using different charging scheduling methods.

(b) Adaptive pricing and NRG-Coin mechanisms.

Fig. 4. Difference in imbalance curves in two different scenarios

Table 2. Pricing Algorithms: Energy differences compared to "no EVs" baseline.

Method	Imbalance	Wasted	Imported	MAPE
NRG-Coin	−31.4%	−45.6%	+16.4%	−44.5%
Adaptive pricing	−31.3%	−45.6%	+17.1%	−42.7%

other two, and the difference (mirroring this *V2G*'s advantage) increases as the duration of connection to a charger gets longer.

The third use case compares different pricing algorithms for the MD agent, in particular the *NRG-Coin* pricing and the *adaptive pricing*. Both methods aim to balance demand and supply, by setting higher prices for consumption during problematic intervals of negative imbalance, and lower for those with positive. The charging of EVs for this use case is performed according to the *Lowest Prices* scheduling approach. Considering that EV agents are rational and aim to reduce their expenses, the application of the two pricing algorithms results to demand being shifted to utilize the generated energy more effectively, thus leading to smaller peaks in the imbalance curve. Figure 4b shows that the algorithms have a similar effect on the stability of the grid. In Table 2, we can observe a similar behavior of reducing the wasted energy and a slightly outperform of *NRG-Coin* on imported energy and MAPE reduction.

In the fourth use case, we count the total number of exchanged messages required for the scheduling of charging using our proposed cooperation protocols as the EV population increases. We report that we observed a *linear* increase in the number of messages exchanged over a 10 days period (we do not present the results in detail due to space restrictions).

6 Conclusions and Future Work

In this paper we presented a open architecture for the V2G/G2V energy transfer problem domain, and provided implementations of agents as flexible microservices that are interconnected by an IoT platform. Our approach can be used for the exploration of various agent strategies in simulation mode, but is also readily deployable and can support real world trials. We also address the needs for openness, and the coverage of diverse business models via the definition of a number of key agent types and the development of open protocols. These can be made available to any interested party, which can subsequently build their own agents given their expertise and business cases. This is demonstrated via presenting realistic use case scenarios.

Having validated our architecture, we can now look to the future. There is much to be done in terms of populating the agents' components with actual machine learning, decision-making, and recommendation algorithms. Finally, we intend to use our system in the real-world, first as part of a pilot study. This will allow us to test the perceived openness and the usability of the system, and to identify potential extensions, as well as important business models.

References

1. Akasiadis, C., Pitsilis, V., Spyropoulos, C.D.: A multi-protocol IoT platform based on open-source frameworks. Sensors **19**(19), 4217 (2019)
2. Burke, M.J., Stephens, J.C.: Energy democracy: goals and policy instruments for sociotechnical transitions. Energy Res. Soc. Sci. **33**, 35–48 (2017)
3. Danner, D., Duschl, W., de Meer, H.: Fair charging service allocation for electric vehicles in the power distribution grid. In: e-Energy 2019, pp. 406–408 (2019)
4. Espe, E., Potdar, V., Chang, E.: Prosumer communities and relationships in smart grids: a literature review, evolution and future directions. Energies **11**(10), 2528 (2018)
5. Hossein Motlagh, N., Mohammadrezaei, M., Hunt, J., Zakeri, B.: Internet of things (IoT) and the energy sector. Energies **13**(2), 494 (2020)
6. International Energy Agency: Global EV outlook: Towards cross-modal electrification (2018)
7. Käbisch, S., Peintner, D., Heuer, J., et al.: The OpenV2G project. http://openv2g.sourceforge.net/. Accessed 22 Apr 2022
8. Karfopoulos, E.L., Hatziargyriou, N.D.: A multi-agent system for controlled charging of a large population of electric vehicles. IEEE Trans. Power Syst. **28**(2), 1196–1204 (2013)
9. Ketter, W., Collins, J., Reddy, P.: Power TAC: a competitive economic simulation of the smart grid. Energy Econ. **39**, 262–270 (2013)
10. Lee, Z., Johansson, D., Low, S.H.: ACN-sim: an open-source simulator for data-driven electric vehicle charging research. In: e-Energy 2019, pp. 411–412. ACM (2019)
11. Liao, J.T., Huang, H.W., Yang, H.T., Li, D.: Decentralized V2G/G2V scheduling of EV charging stations by considering the conversion efficiency of bidirectional chargers. Energies **14**(4), 962 (2021)
12. Lipman, T., Callaway, D., Peffer, T., von Meier, A.: Open-source, open-architecture software platform for plug-in electric vehicle smart charging in California. California Energy Commission (2020)
13. Mahela, O.P., et al.: Comprehensive overview of multi-agent systems for controlling smart grids. CSEE J. Power Energy Syst. (2020)
14. Mihaylov, M., Jurado, S., Avellana, N., et al.: NRGcoin: virtual currency for trading of renewable energy in smart grids. In: 11th International Conference on the EU Energy Market (EEM 2014), pp. 1–6. IEEE (2014)
15. Sarkar, R., Saha, P.K., Mondal, S., Mondal, A.: Intelligent scheduling of V2G, V2V, G2V operations in a smart microgrid. In: e-Energy 2020, pp. 417–418 (2020)
16. Savaglio, C., Ganzha, M., Paprzycki, M., et al.: Agent-based internet of things: state-of-the-art and research challenges. Futur. Gener. Comput. Syst. **102**, 1038–1053 (2020)
17. Sheppard, C., Jenn, A.: Grid-integrated electric mobility model (GEM) v1.0. US Department of Energy (2021). https://www.osti.gov//servlets/purl/1765949
18. Spanoudakis, N., Moraitis, P.: Engineering ambient intelligence systems using agent technology. IEEE Intell. Syst. **30**(3), 60–67 (2015)
19. Valogianni, K., Ketter, W., Collins, J., Zhdanov, D.: Sustainable electric vehicle charging using adaptive pricing. Prod. Oper. Manag. **29**(6), 1550–1572 (2020)
20. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**, 115–152 (1995)
21. Zambonelli, F.: Key abstractions for IoT-oriented software engineering. IEEE Softw. **34**(1), 38–45 (2017)