TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)
7-10 Dec 2021. Auckland, New Zealand

Electric Vehicle Charging Station using Open Charge Point Protocol (OCPP) and oneM2M Platform for Enhanced Functionality

Deeksha Devendra, Shubham Mante, *Student Member, IEEE*, D. Niteesh, Aftab M. Hussain*, *Member, IEEE*
PATRIoT Lab, Center for VLSI and Embedded Systems Technology (CVEST),
International Institute of Information Technology (IIIT) Hyderabad, India
*Email: aftab.hussain@iiit.ac.in

***Abstract*—A smart city is a collection of a variety of smart devices representing several disparate infrastructural verticals. The data gathered by these smart devices helps bring a strong integration of human, collective, and artificial intelligence within the city operation. With the advent and proliferation of electric vehicles (EVs), the EV charge point is a relatively new addition to this plethora of smart devices. We believe that the EV charge point can acquire enhanced functionality if it can be seamlessly integrated with the rest of the smart city infrastructure. For example, charging services can be stopped if smart grid or smart fire alarms sense an electrical overload or fire hazard in the vicinity. However, the interaction of an EV charge point with the smart city infrastructure requires seamless information exchange across various verticals. This horizontal flow of information is enabled by making the charge point complaint with the oneM2M platform. To this end, we present the design and fabrication of an EV charge point based on the OCPP communication standard and complaint with the oneM2M platform. Further, we discuss various use cases showing an increased functional capability of the EV charger due to access to data from other IoT devices.**

***Index Terms*—Electric & Autonomous Vehicles, Intelligent Transportation Systems, Communication Systems, Open Charge Point Protocol(OCPP), oneM2M platform**

I. Introduction

A smart city is a technologically advanced urban area where a wide range of electronic and digital technologies with enhanced functional abilities are used to improve the quality of life of residents based on the data gathered by a variety of sensors. The concept of a smart city depends entirely on smart devices and their ability to communicate data efficiently. Already, advanced sensor systems are being used in various infrastructural establishments such as parking, waste management, traffic regulation, and so on. Electric vehicles and their charging infrastructure is a relatively new entrant in the smart infrastructure domain. Global regulatory agencies and various Governments around the world have set ambitious targets for the adoption of electric vehicles and the development of the charging infrastructure in a hope to combat the ever-increasing threat of human-induced climate change [1].

With a growing population choosing to adopted the wide variety of electric vehicles now available in the market, there is an urgent need to enhance the charging infrastructure. This requires both reliable hardware design and an efficient software implementation to coordinate the charging process. There are already several sets of standards for the fabrication of the charges and vehicles to ensure that every manufacturer is taking care of the safety, security, and efficiency in their device. IEC 62196 standard series defines the general requirements for dedicated plugs, socket outlets, dimensional compatibility and interchangeability requirements for the plug [2]. OCPP standard defines a method enabling charging points to communicate with a central system [3]. JWG1 standard defines vehicle-to-grid communication interface [4]. IEC61851 standard defines different charging topologies needed to be considered for conductive AC and DC based dedicated charging equipment [5]. Levels define the power levels for EV charging station [6], for example, Level 2 uses voltages between 208 and 240 VAC, a capacity of 7.6 kW and a current of up to 32 A.

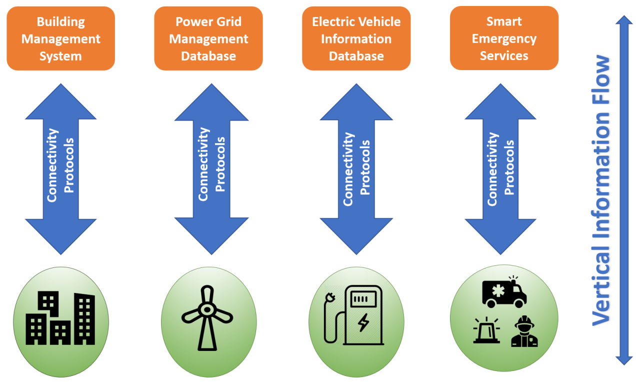

Fig. 1. Vertical flow of information from various infrastructural elements in a smart city.

The methodology involved in building a compact EV charger based on level 2 charging standards for charging E-scooters using the standard open charge point protocol (OCPP) has been discussed in our previous work [7]. The front end of the device consists the user interface, whereas The back end covers the application interface and the client-server interaction of the charger. It is possible to build the EV charger based on the same front end and OCPP protocol with any cloud-based back end. However, an arbitrary cloud-based back end allows only a vertical flow of information from the charging station to the server, which can be insufficient in the context of a smart city deployment.

Figure 1 shows various infrastructural elements in a smart city, such as a smart building, smart grid, EV charger, and smart emergency services. These devices, with the help of connectivity protocols, store and compute the data gathered

TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON) | 978-1-6654-9532-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/TENCON54134.2021.9707311

by their sensors. However, because of vertical flow of data, there is no exchange of information across various segments. For example, a smart grid can only calculate electric power generated and distributed, however, it cannot detect a fire hazard that might be caused due to electrical overloading. Similarly, emergency devices like fire alarms can be used to contact the respective emergency response teams, but cannot be used to cut-off power to an area to avoid a potential fire hazard. These correlations are only possible if the data is seamlessly exchanged between different verticals at the city level and a clear semantic context for the data is available. Thus, the functional abilities of the devices can be enhanced by using the data produced by other sensors. Hence, we require an efficient method for the horizontal flow of information, which can be introduced using OM2M [8], an open-source platform compliant with the oneM2M standard.

The oneM2M platform is similar to a distributed operating system for IoT. It takes the form of a middleware service layer and sits between layers and connectivity transport. OneM2M was initiated as a global partnership project in 2012. The partnership consists of the world's leading countries in information and communication technology standardization development namely, the USA, Japan, China, Europe, India, and Korea. The goal of this organization is to provide a global technical standard for interoperability of machine-to-machine (M2M) technologies by developing standardizations in architecture, API specifications, security and so on. OneM2M currently has more than 200 participating partners which include many leading companies of the world like Samsung, Intel, Adobe, IBM etc. OneM2M has already found its use in many major industries and smart cities. It is of immense use in the health industry to capture, analyze, store, and transmit health information from various sensors and bio-medical acquisition systems [9], [10]. It helps transform buildings into green, energy-efficient buildings by utilizing the collected sensors' data [11], [12]. Moreover, it is used in smart e-bike monitoring system for the real-time acquisition of usage data from electrically-assisted bikes [13].

In this paper, we present the design and fabrication of an EV charger with OCPP as the communication standard and oneM2M platform as the data interfacing and archiving standard. In section 2, we discuss the methodology to build an EV charger hardware and software stack following the OCPP and oneM2M standards. We have divided the process into three subsections: the authentication process, the billing process and the status update process. In section 3, we outline the property of horizontal flow of information introduced in EV due to oneM2M platform and conclude with a discussion of the use cases of the above property.

II. EV Charger on the oneM2M Platform

The details of the hardware design of the EV charger have been presented in our previous work (Figure 2) [7]. The dimensions of the enclosure are 22 cm x 22 cm x 6 cm. It has a 7-inch LCD touch screen display at the front and a 3-pin output plug at the side. In this design, an AC supply is first connected to the emergency stop button and then goes through the power meter and relay module to the output plug. The AC power is used as the input for powering the processor and other electronic components. The user interacts with the charger through the RFID reader, input keypad and the LCD touch screen.

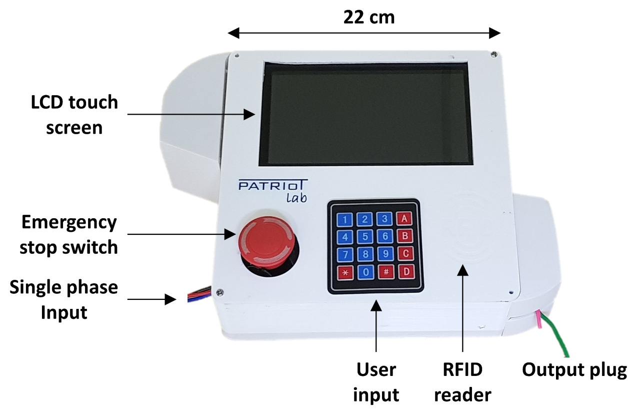

Fig. 2. The EV charger has been designed to work on 220 V, single phase AC supply, and is compatible with an electric scooter/bike from any manufacturer.

The user is greeted with a welcome screen with a tab to initiate the diagnostic process of the charger. This step takes a few seconds and ensures that the charger is in a working state and ready for charging. Once the user is notified that the charger is in the operative state, he/she proceeds to authentication. The user swipes the RFID card provided by the service provider or scans the QR code from a smartphone. The user then enters the amount for which the charging has to be done. The user authentication information is sent to the server which confirms the existence of the user profile and fetches the balance in the e-wallet present for the specific user and sends the balance as a secure data packet to the client. If the authentication is not approved, the charger defaults back to the welcome screen. Once the authorization is completed successfully, the system notifies the user that it is ready for charging. The user then confirms to begin the charging process, and the power supply to the electric vehicle is initiated. During the charging process, the charger continuously monitors the rate of charging, and notifies the user of the time left for the completion of the charging process. At the end of charging, a thank you screen is displayed. In case the charging stops before the targeted amount are reached, the unused amount is added to the user's

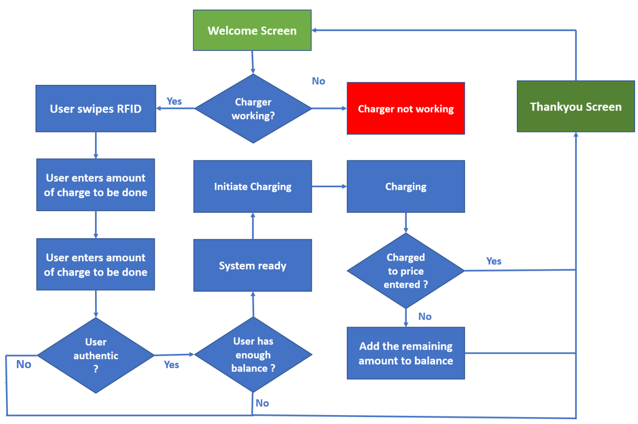

Fig. 3. The user experience flow for the designed EV charger.

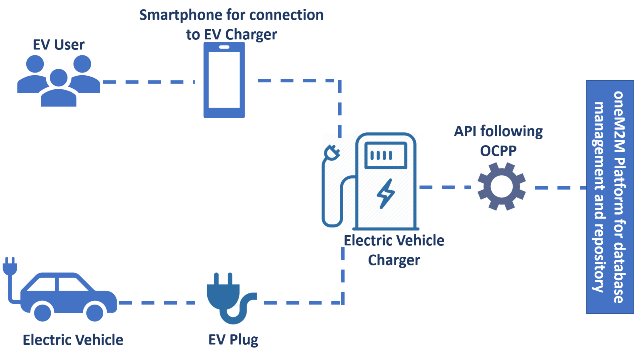

Fig. 4. The workflow of the software application of the EV charger.

balance in the e-wallet. The user experience flow is presented in the flowchart in Figure 3.

Now, let us consider the implementation of these steps using OCPP and oneM2M standards. Figure 4 summarizes the entire charging workflow from the eyes of software development. The complete charging flow works on the directives provided by the Open Charge Point Protocol (OCPP), an initiative by Open Charging Alliance (OCA), an international affiliate organization [14]. OCPP is an application protocol that establishes communication between the charging system (charge point) and the central management system (CMS) [15], [16]. The CMS is the cloud-based back end system that operates the charging process [17]. In this work, we propose a novel approach to create the CMS based on the oneM2M platform. Thus, OCPP is used to interface between the oneM2M platform and the charge point. Due to the interfacing, the charger becomes an oneM2M complaint device independent of the device technology. The oneM2M platform then provides application developers with a standardized set of APIs for uniform monitoring or control of the charger along with other IoT devices of the entire smart city implementation, which enables seamless data flow between different verticals. The entire charging mechanism is classified into three parts: the authentication process, the billing process, and the status update on the cloud after the charging.

A. Authentication Process

Figure 5 represents the transfer of information from a non-oneM2M node (node that does not contain oneM2M complaint entities, example, EV charger) to the oneM2M service layer through an interworking proxy entity (IPE). The authentication process involves transferring data in JSON (JavaScript Object Notation) format from the charge point (or charging system) to the oneM2M platform and vice-versa. The authentication process begins with verifying whether the user is already registered in the system or not. Our EV charger uses RFID for user authentication, which is received when the user tries to charge the vehicle using an RFID tag. As soon as the charger gets the RFID, it sends a GET request using REST APIs provided by the oneM2M standard to the platform to retrieve the information associated with that RFID. If the user is not registered, the platform sends an error message. A message saying “user not found” will be displayed on the charger screen, and therefore, the user must first register to use the charger. In the case of successful user authentication, the charger compares the user’s account balance, and if it is greater than the required amount, the user can charge the vehicle; otherwise, an insufficient amount in the account message will be displayed.

B. Billing Process

In case of successful authentication, the balance from the user’s e-wallet and cost of per unit charge of consumption is retrieved from the cloud. The data is received in JSON format. This has been implemented in the EV charger (client) using Python 3. We needed to deserialize the delivered JSON data into a Python object. The deserialization of data was done with the help of method $loads()$ available in Python 3. Once we get access to the balance in float datatype, we can use it for computation. The computations are done on the end device in order to eliminate latency and network related issues. It compares the amount entered by the user to the balance fetched from the account. If the balance is sufficient, the charger proceeds to notify the user the amount that the charging will be done for and the estimated time it will take to complete the charging process. After the charging stops, a variable named $extra$ is assigned to check whether the charging is done for the entire amount. In case the charging stops abruptly due to some emergency or electric power failure, the processor of the charging system works on an in-built battery backup to update the server about the amount of the charging completed. When the charging is done for the amount targeted by the customer, $extra$ is updated to 0.

C. Status update on cloud

Figure 6 shows the structure of the EV charger application entity (AE) on the oneM2M platform, say as one of the AEs of a smart building resource tree. In the first layer of AE,

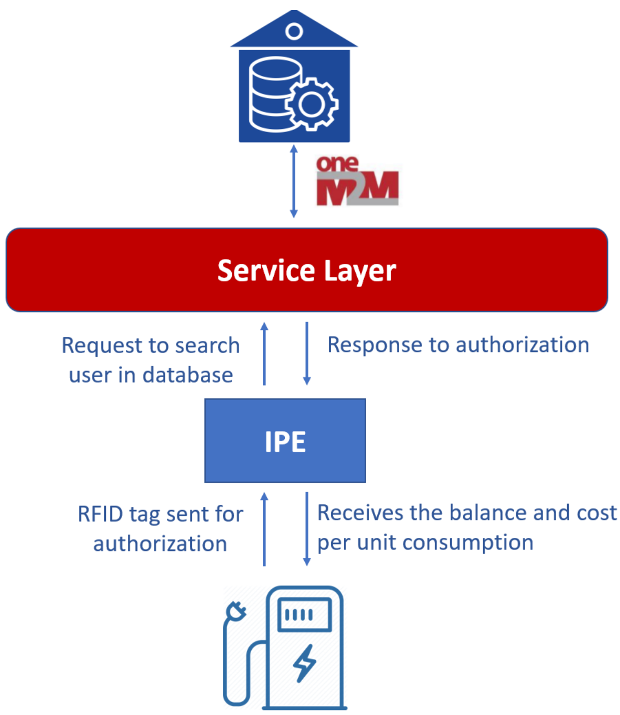

Fig. 5. The authentication workflow from the EV charger to the oneM2M platform.

two containers are used to separate user and charge point information. The second layer is created to add the users' and chargers' identification information. A container for the registered user will be already created under the USER-DATA container during user registration. This contains the user ID information fetched during the authentication process. For an unregistered user, if the charger tries to fetch the ID, the platform throws an error that the user has not been found. In the final layer, user and charger information will be stored along with the transaction history. Therefore, suppose a user finishes charging, then the extra amount is added to the balance (which is now equal to the balance before the charging amount the user entered for the charging). The updated balance is sent to the platform and updated in the user and charger data's transactions containers by creating a new content instance. The charger will then use this information during the next transaction cycle to check if the user has enough balance to charge the vehicle.

III. Interaction between EV Charger and Other End-devices

Modern devices have a set of protocols defined for the flow of information among different layers of communication. oneM2M platform enables communication among multiple IoT domains thereby permitting the horizontal flow of information. It provides a service layer or middleware that sits between the applications and the underlying processing or communication hardware. These services are accessed through uniform APIs. These APIs provide a set of defined tools, definitions and protocols that are applicable to all IoT devices. This set of common functionalities are called common service functions. There are a variety of common service functions like data management and repository, subscription and notification, device management etc. [18]. In this section, we will consider two use-cases for using oneM2M platform in the EV charging infrastructure. These include the information exchange between the charge point and smart emergency devices like fire-alarms in the buildings, or with a smart grid. We begin by reviewing the five layers of IoT architecture and classifying the different processes involved in communication among smart devices into these layers. The sensing layer or the perception layer is the physical layer where the variety of sensors in the devices gather information from the end-user. The EV charger, smart grid, smart building or infrastructure, and, smart emergency devices come in this segment. Once the data is gathered we need to transmit the data to the server for required computation and archiving. This happens in the network layer and the transmission of data is made possible by a variety of connectivity protocols like Bluetooth, LoRaWAN, Wi-SUN, LAN, Wi-Fi, cellular data, etc. The data then reaches the middle service layer. This layer helps store, and process the massive amount of data that comes from the network layer. In this work, oneM2M acts as a middle layer for storing data from the smart devices (EV Charger, grid etc.). The application layer is responsible for delivering specific information asked by the user. Out of a variety of data registered by the sensors, the user typically needs a small subset of very specific data which is fetched and displayed by a dashboarding application. The business layer includes the various models for computation, information analysis, users' security etc. [19].

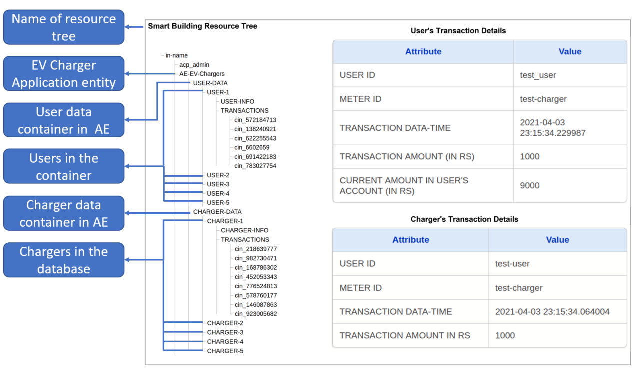

Fig. 6. Screenshot of the EV charger application entity implemented on the open source OM2M platform.

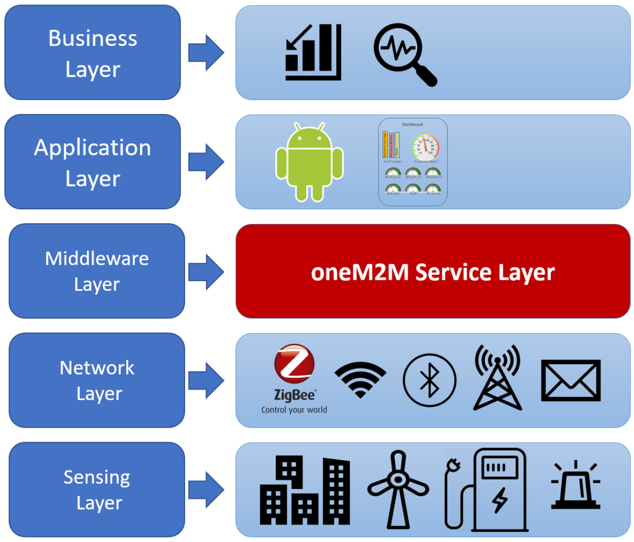

Fig. 7. Different layers of IoT architecture and the placement of oneM2M platform as the middle layer.

A. *Interaction between EV Charger and smart sensors*

One of the most used smart emergency devices is a fire sensor. This module deals with interaction between two application entities (AEs), the EV charger and the fire alarm sensor residing under the same common service entity (CSE). The sensor helps detect smoke or fire-related emergencies. Because EV charging is a high-power application, it needs to be immediately terminated in case of fire in a building. The data regarding activation of the fire alarm in the building is set as a boolean variable on the oneM2M platform. During an emergency, the alarm sets the state of this variable to true. As soon as the alarm notifies the oneM2M service layer of its activation state, an actuation command is sent as a server sent event (SSE) to the EV charge point to turn it off. The transaction history is created on oneM2M and the remaining balance is updated in the user profile. Such a cross-vertical application is not possible if the EV charging system and the fire alarm system operate independently. The use of the oneM2M platform as the middleware layer enables the seamless exchange of information.

B. Interaction between EV Charger and smart grid

Smart-grid is an electrical grid that equipped with digital sensors to detect and monitor the generation, transmission and distribution of electrical energy. Some of the basic features include load adjustment or load balancing, an account of power utilized in a specific duration of time, automatic fault detection, automatic theft detection and so on. It helps in the reduction in line-losses on both transmission and distribution sides. The total load connected to the power grid varies significantly over time based on the time of day and demand surges. With such a system in place, the smart grid can notify the oneM2M platform of the specific time during which overloading is expected. This can produce an actuation command that is sent by the platform notifying the charge point it is not safe to provide service at the given point in time. Further, the charge point can communicate with the smart grid to notify the user if there is differential pricing for charging during expected surge times. This can help balance the demand from charging infrastructure with the rest of the grid. The

IV. Conclusion

In this paper, we have described the design and fabrication of an EV charge point using OCPP protocol with oneM2M as the service layer. Such an implementation enables the charge point to integrate and exchange information with the smart city infrastructure. The concept of smart city becomes truly successful only when more number of devices interact with each other and utilize the data gathered by other devices. We have focused our discussion on using oneM2M as the cloud based back end for the OCPP protocol. We have also summarized some usecases that are possible only because of the interaction of the EV charger with different smart devices and showed how oneM2M can enable data exchange across verticals. Our future course of action is the development of a highly scalable system that can handle millions of charge points across the city along with data from all the various infrastructure elements in the smart city.

Acknowledgment

This research was partly supported by the Ministry of Electronics and Information Technology (MEITY), Government of India, under grant no. 3070665 (2020) as part of the Smart City Living Lab project. We also thank IHub-Data, IIIT Hyderabad for their support.

References

[1] A. M. Omer, "Energy, environment and sustainable development," *Renewable and sustainable energy reviews*, vol. 12, no. 9, pp. 2265–2300, 2008.

[2] A. Khaligh and M. D'Antonio, "Global trends in high-power on-board chargers for electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3306–3324, 2019.

[3] Á. Rodríguez-Serrano, A. Torralba, E. Rodríguez-Valencia, and J. Tarifa-Galisteo, "A communication system from EV to EV Service Provider based on OCPP over a wireless network," in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 5434–5438.

[4] S. Käbisch, A. Schmitt, M. Winter, and J. Heuer, "Interconnections and communications of electric vehicles and smart grids," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 161–166.

[5] A. Di Giorgio, F. Liberati, and S. Canale, "IEC 61851 compliant electric vehicle charging control in smartgrids," in *21st Mediterranean Conference on Control and Automation*. IEEE, 2013, pp. 1329–1335.

[6] L. González, E. Siavichay, and J. Espinoza, "Impact of EV fast charging stations on the power distribution network of a Latin American intermediate city," *Renewable and Sustainable Energy Reviews*, vol. 107, pp. 309–318, 2019.

[7] D. Devendra, S. Malkurthi, A. Navnit, and A. M. Hussain, "Compact electric vehicle charging station using open charge point protocol (OCPP) for e-scooters," in *2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET)*. IEEE, 2021, pp. 1–5.

[8] M. B. Alaya, Y. Banouar, T. Monteil, C. Chassot, and K. Drira, "OM2M: Extensible ETSI-compliant M2M service platform with self-configuration capability," *Procedia Computer Science*, vol. 32, pp. 1079–1086, 2014.

[9] C. Thuemmler, "The case for health 4.0," in *Health 4.0: how virtualization and big data are revolutionizing healthcare*. Springer, 2017, pp. 1–22.

[10] W. D. de Mattos and P. R. Gondim, "M-health solutions using 5g networks and m2m communications," *IT Professional*, vol. 18, no. 3, pp. 24–29, 2016.

[11] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings," *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

[12] P. Bellagente, P. Ferrari, A. Flammini, and S. Rinaldi, "Adopting IoT framework for Energy Management of Smart Building: A real test-case," in *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. IEEE, 2015, pp. 138–143.

[13] C. Kiefer and F. Behrendt, "Smart e-bike monitoring system: real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles," *IET Intelligent Transport Systems*, vol. 10, no. 2, pp. 79–88, 2016.

[14] S. Orcioni, L. Buccolini, A. Ricci, and M. Conti, "Electric vehicles charging reservation based on OCPP," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2018, pp. 1–6.

[15] C. Alcaraz, J. Lopez, and S. Wolthusen, "OCPP protocol: Security threats and challenges," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2452–2459, 2017.

[16] S. Ravindran, S. Amal, Y. Bhavya, and V. Chandrasekar, "OCPP based Electric Vehicle Supply Equipment and its user interface for AC charging in Indian scenario," in *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE, 2020, pp. 1–6.

[17] T. V. Pruthvi, N. Dutta, P. B. Bobba, and B. S. Vasudeva, "Implementation of OCPP protocol for electric vehicle applications," in *E3S Web of Conferences*, vol. 87. EDP Sciences, 2019, p. 01008.

[18] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[19] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures," *Protocols, and Applications.*