electronics

MDPI

Article

General Overview and Proof of Concept of a Smart Home Energy Management System Architecture

Lucas L. Motta [1], Luiz C. B. C. Ferreira [1], Thales W. Cabral [1], Dimas A. M. Lemes [1], Gustavo dos S. Cardoso [1], Andreza Borchardt [1], Paulo Cardieri [1], Gustavo Fraidenraich [1], Eduardo R. de Lima [2], Fernando B. Neto [3] and Luís G. P. Meloni [1,*]

[1] Department of Communications, School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-852, Brazil; l192701@dac.unicamp.br (L.L.M.); l144401@dac.unicamp.br (L.C.B.C.F.); t264377@dac.unicamp.br (T.W.C.); diguto@decom.fee.unicamp.br (D.A.M.L.); g264375@dac.unicamp.br (G.d.S.C.); a264367@dac.unicamp.br (A.B.); cardieri@unicamp.br (P.C.); gfraiden@unicamp.br (G.F.)
[2] Department of Hardware Design, Instituto de Pesquisa Eldorado, Campinas13083-898, Brazil; eduardo.lima@eldorado.org.br
[3] Companhia Paranaense de Energia, Curitiba 81200-240, Brazil; fernando.bauer@copel.com
* Correspondence: meloni@unicamp.br; Tel.: +55-19-3521-3830 (ext. 3818)

Abstract: This paper proposes and implements a smart architecture for Home Energy Management Systems (HEMS) that enables interoperability among devices from different manufacturers. This is achieved through the use of standardized elements and the design of an innovative middleware. The system comprises a control unit that communicates with smart outlets using the Wireless Smart Ubiquitous Network (WI-SUN) Home Area Network (HAN) specification, while smart metering is achieved using the WI-SUN Field Area Network (FAN) specification. To manage important data, a web platform and mobile app were created. Additionally, machine learning techniques are utilized to identify energy consumption of individual appliances when only the aggregate energy consumption of the house is available. The architecture presented here supports real-time control of energy use and generation through HEMS, and new devices can be added transparently. Finally, a comparison of the proposed system with similar systems in literature highlights its many advantages in terms of functionality.

Keywords: HEMS; IoT; open middleware; cloud processing; big data; machine learning

check for updates

Citation: Motta, L.L.; Ferreira, L.C.B.C.; Cabral, T.W.; Lemes, D.A.M.; Cardoso, G.d.S.; Borchardt, A.; Cardieri, P.; Fraidenraich, G.; de Lima, E.R.; Neto, F.B.; et al. General Overview and Proof of Concept of a Smart Home Energy Management System Architecture. *Electronics* **2023**, *12*, 4453. https://doi.org/10.3390/electronics12214453

Academic Editor: Nikolay Hinov

Received: 19 September 2023
Revised: 18 October 2023
Accepted: 19 October 2023
Published: 29 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

One important challenge that all countries currently face is balancing the growing demand for energy with the need to address sustainability requirements that can reduce the environmental impacts of energy generation. Electrical energy is a scarce resource, making the management of consumption by homes and companies crucial for ecological reasons and essential for allowing power utilities to efficiently meet demand.

Home Energy Management Systems (HEMS) represent a key element in efforts to use electrical energy more efficiently. These systems can be installed in households and commercial buildings to monitor the energy consumption of appliances such as air conditioners, refrigerators, and microwaves. The collected energy consumption data can then be used along with tariff plans offered by energy providers (typically with rates varying with the period of the day) to implement energy-saving strategies such as scheduling the use of certain appliances when tariffs are low. Therefore, the HEMS system is relevant for the consumer electronics segment, as it provides users with improved control over energy consumption. It avoids unnecessary expenses by providing information on preventive and corrective maintenance and control of devices, thereby minimizing energy waste.

The measurement of energy consumption and on/off state control of appliances through HEMS is usually achieved intrusively using so-called *smart outlets*, which communicate with a central device called an *HEMS controller*. The use of smart outlets allows the energy consumption of any appliance to be monitored without requiring any special function in the appliance itself. In the future, energy measurement and communication protocols could be embedded in the household appliances themselves, which would require standardized data structure and communication protocols. Although there is no de facto standardization, protocols such as the Wireless Smart Ubiquitous Network (Wi-SUN) standard enable manufacturers to offer interoperable products [1]. The WI-SUN standard is a consolidated communication specification established by the WI-SUN Alliance, and includes more than one hundred member companies. WI-SUN is suitable for employment in wireless networks with low data rates, high coverage areas, and low power consumption [2].

The use of a communication layer to work with different communication protocols through standardized middleware greatly facilitates HEMS interoperability among devices. Such a middleware can make operation with new devices easier than using the definition in the communication protocol, allowing new applications to run in the controller or the cloud. The middleware can be required to mediate interactions between applications and hardware as well as among systems. Without these solutions, programmers would have to read a new software specification every time they integrate new packages, making these tasks difficult and very time-consuming.

In certain scenarios, houses using HEMS may not have a smart outlet dedicated to each single appliance. In such cases, a smart meter installed at the house's energy input may collect the total energy consumption of all appliances in the house. The consumption of each appliance can be estimated using load disaggregation techniques, which can be implemented using machine learning models trained on data collected from other houses and similar appliances.

In this work, we present the design and implementation of an HEMS system that allows interoperability among devices from different manufacturers by adopting standardized elements. In addition, we propose an open and innovative middleware. The system consists of a control unit and smart outlets that communicate through the WI-SUN Home Area Network (HAN) specification. The controller communicates with the smart meter using the WI-SUN Field Area Network (FAN) specification. The middleware is part of this architecture, providing an application connectivity layer between the software elements and the hardware resources without requiring customized programs. In the proposed system, users can visualize the energy consumption data and statistics of different appliances. In situations where only aggregated energy measurements are available, machine learning techniques are employed to distinguish the energy consumption data of the different appliances in the house using load disaggregation techniques.

The proposed architecture represents an alternative for resolving interoperability problems between communication standards and services, including hardware solutions and software, for both users and electrical utilities. In addition, the proposed system incorporates the possibility of collecting and processing further data, which can then be used to provide new functionalities and services.

Major Contributions

This paper introduces the design, implementation of Proof of Concept (PoC), and testing of a complete home energy management system. The proposed contribution consists of an entire system, from smart outlets to web and mobile applications.

According to our literature survey, various communication protocols for wireless sensor networks (WSN) are available for Home Energy Management Systems (HEMS). Our proposal involves a flexible hardware and open middleware architecture that can integrate different devices, enabling the HEMS to handle numerous communication protocols. This

middleware architecture provides a user friendly approach for developing new applications using a microservices architecture.

The design of our proposed HEMS includes an AI-based load disaggregation technique that can be applied to households lacking smart outlets where a smart meter collects consumption data from all appliances. The HEMS is designed with User Interface/User Experience Design (UI/UX) web and mobile applications that display the contribution of each appliance to the overall household electricity consumption. Additionally, we have included a web application for energy providers that presents information regarding the energy consumption in a particular region.

The primary contribution of our work is a modern and flexible HEMS architecture that can provide essential services and potential future applications. The major contributions of this work are:

- A proposed middleware architecture focused on HEMS that uses a microservices design while having a light footprint and low computational complexity, and which can be easily customized by automatically creating and deploying new services.
- Regarding the wireless HAN specification, the ability to choose between several different technologies makes the system more flexible compared with other similar systems.
- To the best of our knowledge, this is the first paper to present a complete system for an HEMS controller confirming to the WSN Wi-SUN world standard specification for smart meters.
- The solution presented in this work is flexible regarding hardware architecture, allowing daughter boards for CPU modules and wireless interfaces using inexpensive hardware platforms, which is feasible thanks to the characteristics middleware.
- This work addresses issues related to cloud software, web applications, and security, which have not been discussed collectively with respect to other systems in the literature.
- A mobile app running on Android and iOS devices provides users with a detailed history of their power consumption.
- The system employs a state-of-art disaggregation technique based on machine learning algorithms [3] that runs as a middleware application.
- The overall architecture is designed to be flexible in order to cover relevant aspects of possible future scenarios around home energy management systems. This includes the use of AI-based applications and big data techniques, which could be built on top of the proposed architecture.

The proposed system outperforms similar works found in the literature [4–9] in terms of many important characteristics and functionalities, as presented in Table 1. Regarding the wireless protocol, the use of different technologies allows the system to be more flexible than others, as it is the only system to employ a communication protocol with the smart meter based on the Wi-SUN FAN standard. Our work provides an open middleware with low computational power consumption that covers all the resources needed for HEMS applications. In addition, the proposed system employs modern microservices concepts, and can be easily customized via automatic development and deployment of new services.

In terms of hardware components, the proposed system offers higher processing and memory capacity than other systems despite utilizing affordable hardware platforms. This is possible due to the unique characteristics of the developed middleware. Unlike other works, our proposal addresses includes built-in web application, security, and mobile app functionality.

The mobile app provides consumers with detailed billing information, allowing for better cost control, and is compatible with both Android and iOS operating systems, providing greater flexibility. Additionally, our proposal includes a machine learning solution developed as a middleware application to performs the load disaggregation task with low computational complexity and high performance. This enables the estimation of information in residences where smart outlets are unavailable.

Table 1. Comparison between our proposed system and other similar systems found in the literature.

	Wireless HAN Specification	Wireless FAN Specification	Middleware	Hardware	Security	Web App	Mobile App	Machine Learning Algorithms
Our Proposed System	Wi-SUN HAN/ Wi-Fi/ Diverse Protocols	WI-SUN FAN	Yes (Open, Light Footprint, Microservices Architecture, Edge Approaches)	Smart Outlet: Low Power MCU 16 bits 12 MHz, Power Measurement by Metrology FW Inside MCU and HEMS-Controller: NXP iMX6 32 bit ARM MCU	Yes	**Yes**	Yes	Yes
Han et al. [4]	ZigBee	No	No	Not specified	No	Yes	No	No
Han et al. [5]	ZigBee	ZigBee	No	Not specified	No	Yes	No	No
Al-Ali et al. [6]	Wi-Fi	No	Yes (Several Software Tools, MQTT Broker, Cloud Storage, uses high computacional capabilities, cloud approach)	The hardware consists of a sensor array, high-end microcontroller (1MB Flash, 128 kB RAM), and relay banks.	Yes	No	Yes (However, the app does not inform the individual bill for each appliance)	No
Zhou et al. [7]	ZigBee	None	None	RS485	None	None	None	Yes
Shareef et al. [8]	ZigBee	ZigBee	None	XBee S2C Pro IR transmitter TSOP38238	None	None	None	None
Pawar et al. [9]	ZigBee	ZigBee	None	Smart Outlets: Atmega 328 SEM Unit: Microcontroler with Ethernet Shield	Yes	Yes	None	Yes

Hence, our solution offers advantages with respect to most of the issues listed in Table 1. Notably, this work introduces novelties concerning elements of the development and integration of the components of HEMS into a complete solution. For example, load disaggregation (Section 4.2) was designed specifically for this work. Other innovative aspects include the use of microservices to provide the functions (Sections 3.2 and 3.3), the edge computing approach employed by the middleware (Section 3.2), and the use of modern hardware (Section 3.1).

The rest of this paper is organized as follows. Section 2 reviews the literature, focusing on relevant aspects of middleware for Internet of Things (IoT), management systems, machine learning techniques, and big data techniques for HEMS. Next, Section 3 introduces the Smart HEMS Architecture and Middleware (SHArM), which is a modern architecture conceived to be ready for future trends and evolution while allowing upgrades and deployment of new protocols through flexible hardware platforms and an innovative open middleware. Section 4 presents the results of our proof of concept tests on the proposed HEMS architecture, along with a state-of-the-art load disaggregation technique presented as a middleware application. In addition, this section presents details of the software architecture for cloud servers and mobile apps for local and remote accesses. Finally, Section 5 concludes the paper and presents possible future extensions.

2. Literature Review

Our literature review showed a wide range of works on energy management systems (EMS), including surveys of the state-of-the-art [10–14]. In the choice of works presented in this literature review, we favored the following criteria: shortest time since publication, highest publication impact factor, largest number of citations, and best-known authors, in which we followed [15].

The first conceived EMS represented a significant contribution to address the energy crisis of the 1970s. The first work presented in the technical literature dates back 1982,

where an energy management systems using a microcomputer was proposed [16]. Early operations involved analog meters, third-generation computers, and the first commercial microprocessors [17,18]. The arrival of personal computers in the 1980s impacted the EMS field, allowing the development of optimization algorithms for energy management to reduce both electricity cost and demand [16].

In 2003, Kushiro et al. [19] and Inoue et al. [20] presented the design and implementation of a home energy management system composed of a gateway controller and a network architecture for appliance nodes. These gateway/controller devices were installed in homes; they communicated with home appliances and power meters through a power line communication (PLC) system and with a central management system through the internet. This central management system analyzed the collected information and generated operation plans using weather information.

In 2011, Han et al. [4] designed and implemented an HEMS in which home appliances and room lighting could be controlled locally by an infrared remote control or by a web application. The proposed system included dimmable lights, smart outlets integrated with Zigbee modules, and a ZigBee hub. A home server communicated with each ZigBee hub installed in rooms and sent collected data to the internet. The web application enabled users to view home energy data and control home devices remotely through the internet. The UI displayed the energy usage of each home device at different time scales. An intelligent service that automatically turned off home appliances on standby was proposed as well. Years later, in a similar HEMS project, Han et al. [5] included a PLC-based gateway to monitor the generation of renewable energies. The home server gathered energy consumption data, analyzed them for energy estimation, and controlled a home energy use schedule to minimize energy costs. On the other side, the remote energy management server aggregated the energy data to generate statistics.

In 2014, Zhou et al. [7] proposed an HEMS for real-time control of devices, including water heaters, air conditioners, clothes dryers, electric vehicles, photovoltaic cells, critical loads, and battery systems. The data were initially obtained from simulation and later through laboratory tests using hardware for measurements. Using these data, the authors investigated demand response issues via optimization and prediction. An intelligent system for charging and discharging batteries based on fuzzy logic systems was presented, achieving a good performance for the demand response problem.

In 2017, Al-Ali et al. [6] designed and prototyped an HEMS connecting home appliances, heating, ventilation, and air conditioning (HVAC) units to monitoring and control modules with a unique IP address. These modules collected energy consumption data from each smart home appliance and transmitted them to a centralized server using the Message Queue Telemetry Transport (MQTT) protocol and WiFi connection. A central server handled the data from all residential areas for further processing. The proposed EMS used off-the-shelf business intelligence and big data analytics software packages to manage energy consumption and meet consumer demand. The results of analysis using these data can empower local and regional utilities to identify energy consumption patterns according to their respective privileges.

In 2020, Shareef et al. [8] implemented a system employing multiple smart outlets, monitoring nodes, and a remote control for air conditioners. Based on the readings of the outlets, the system uses ON/OFF status for the equipment and measures energy consumption. On the other hand, the nodes provide room monitoring and collect comfort data. Furthermore, the authors employed a central controller algorithm based on rules to balance comfort and energy expenditure, aiming to reduce energy consumption and daily cost. Communication among the devices is based on the ZigBee protocol. The authors demonstrated that the proposed system can reduce daily energy consumption.

In [9], a system incorporating energy management and renewable sources was proposed. Devices such as lighting, fans, and a laptop computer constituted the environment of the study, all of which were connected to a smart outlet. The outlets communicated via ZigBee with a central unit called the Smart Energy Management (SEM) unit, which was

responsible for trading the energy used in the loads. Based on the predicted photovoltaic generation power, the SEM unit transmits commands to the smart outlets and performs management, for which it employs machine learning algorithms such as Artificial Neural Network (ANN), Support Vector Machine (SVM), and Decision Tree (DT). Furthermore, the algorithms address issues related to photovoltaic generation, such as solar irradiation and weather forecasting. A web page for monitoring energy consumption and additional parameters is available to the user as well.

In 2023, the authors of [21] conducted a review of concepts and architectures for Home Energy Management Systems (HEMS) spanning the period from 2015 to 2022. Their review analyzed various characteristics, such as architectures, control center models, topologies, communication standards, and machine learning techniques, among others. None of the works surveyed in this review employed the Wi-SUN communication standard or used a middleware to address the device interoperability problem. Furthermore, the machine learning techniques used in our work are more recent and innovative, representing the state of the art in this type of application. As a result, the advantages and innovations presented in the current study are evident.

At present there are a significant number of middleware platforms available for IoT. In [22], a survey of middleware propositions showed that most solutions are based on the concept of platform as a service, in which the middleware is implemented in the cloud. In the context of HEMS, a number of works in the literature have used middleware for energy management applications [13,23–25]. However, all of these implementations rely solely on cloud computing, and none are adapted for inexpensive hardware with restricted computing capacity.

Current HEMS solutions for home comfort and energy savings need to deal with large amounts of data. AI-based methods such as Machine Learning (ML) are relevant for capturing user behavior, identifying patterns, and simplifying data analysis. In [26], the authors addressed the big data paradigm in HEMS-IoT with the assistance of ML, for example, the C4.5 decision tree algorithm. The ML algorithm is used to help the system with energy-saving recommendations to the user. In [27], the authors used well-known algorithms such as AdaBoost, C4.5, and random forest. In [28], the authors introduced an IoT-based smart home control system implementation incorporating an SVM able to distinguish residents from intruders.

3. SHArM: A Smart HEMS Architecture and Middleware

The innovative, modern, and flexible platform defined in this work is ready for the future, meeting the requirements and challenges of smart home energy systems.

The challenges in designing such an architecture are enormous, including:

- The ability to preview demand-side management (DSM) with appropriate application interfaces for both consumer incentives and distribution policies. DSM allows planning, implementation, and monitoring activities designed to influence consumer energy use in ways that will produce desired changes in the shape of the distributor's load, e.g., peak clipping, load shifting, valley filling, strategic conservation, strategic load growth, and flexible load shape. The proposed system must allow such goals as consumer incentives while respecting their privacy and free will.
- Providing a renewable energy management system integrated with IoT sensors, actuators, and the Smart Grid (SG), allowing information exchange between households and utility to optimize energy consumption, costs, quality of service, and other metrics. In the next decades, solar panels and other renewable energy sources will become more pervasive; thus, prosumers' incentives must preview win–win scenarios.
- Allowing the intensive use of social networks and app development, where the anonymous sharing of data and use of intelligent applications must provide energy cost savings. This environment must preview minimal consumer interventions.

A typical HEMS offers five essential services: monitoring, logging, control, management, and alarms [13]. The proposed HEMS was designed to handle these and other new services based on the architecture shown in Figure 1.

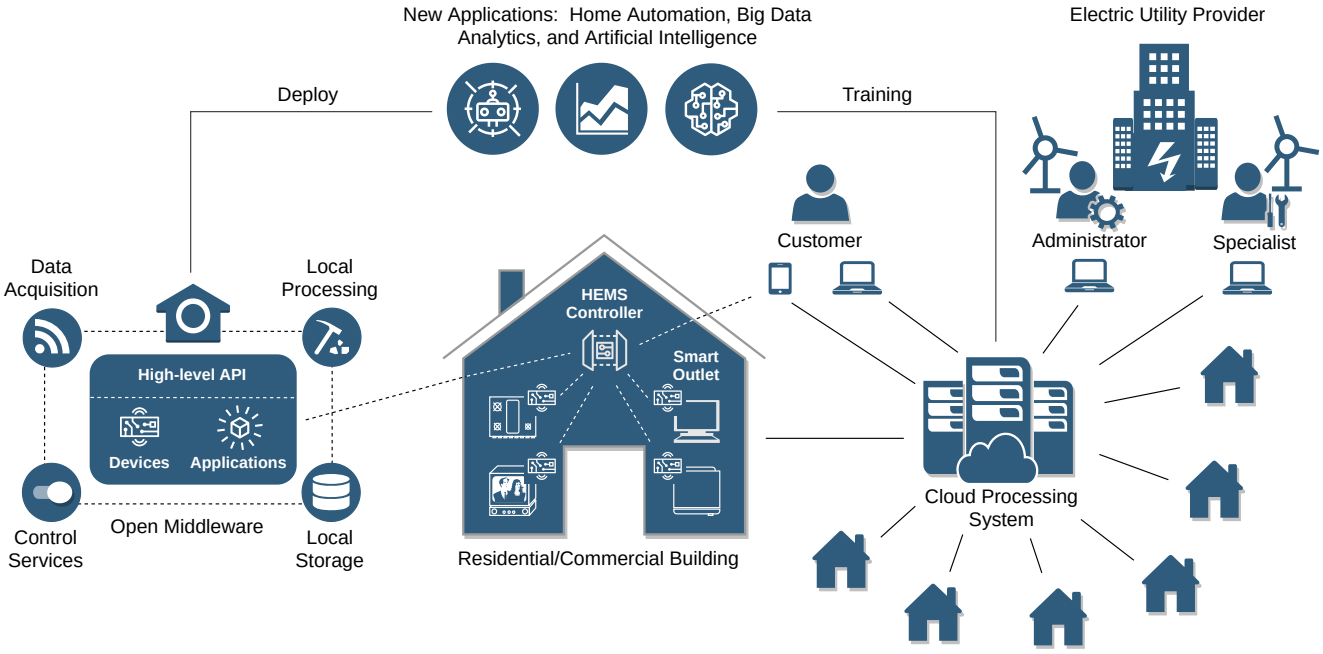


Figure 1. Illustration of the overall system architecture.

Residential and commercial buildings use a set of smart outlets and an HEMS controller, with appliances and other electrical devices are connected directly to smart outlets to collect energy consumption data and turn devices on/off. These smart outlets are in continuous communication with the controller in order to send collected data and receive commands. The controller preprocesses the collected data and sends the data to the cloud servers over the internet. The proposed open middleware embedded into the controllers greatly facilitates communications and promotes interoperability between devices. All data coming from the controllers are indexed and stored in databases on cloud servers. These data remain available for other services and applications. The system includes a web application and a cross-platform mobile application, allowing customers to visualize graphs and statistics on energy consumption and to control the appliances within their buildings and businesses. The mobile application allows access to local (including offline from external access) or remote (through the internet) databases. In addition, the web application has additional services dedicated to the energy utility provider, as consumption statistics are drawn from the data collected by the various controllers.

The proposed architecture includes a middleware that operates close to the end user according to the edge computing paradigm. In this approach, the control functions related to the end user are close by, avoiding delays and unavailability of external connections. Furthermore, our solution periodically sends information to the cloud system for further processing and use. In this context, the architecture has a feature for temporarily storing data in case of internet connection loss. Hence, the architecture uses an approach based on both edge and cloud computing, providing balance and enhanced performance of the HEMS system.

3.1. Flexible Hardware Architecture

A key requirement of our project is to use hardware components that are easy to install in consumers' premises while being suitable for the designs and building materials of most houses. The radio module connects devices using non-licensed ISM frequency bands at 900 MHz or 2400 MHz, which are very well suited for this application. Thanks to the

growth of the IoT sector, there are excellent components and radio devices available for ISM bands at prices low enough for use in cheap products.

While the power consumption of the smart outlet itself is not a significant factor in the design, it is desirable to keep it as low as possible in order to meet the design goals. We assume that all equipment can extract energy from the mains on monitoring and that it will be in operation only when energy consumption is available; therefore, an uninterruptible power supply is not needed.

Another aspect of paramount importance for system implementation concerns the security of the communication process, as all communication devices are susceptible to hacking attacks. The radio technologies described below are already adequate for this purpose. Many other technologies based on star or mesh topologies are available in the market, several of which are open, while others are based on proprietary technology.

3.1.1. HEMS Controller

The HEMS controller is the device that collects the energy data from the smart outlets. It has an LCD touch screen to show user data consumption and to configure the operation and the outlet pairing procedure. All the components are packaged in a case that can be fixed onto walls.

Figure 2 shows the block diagram of the HEMS controller.

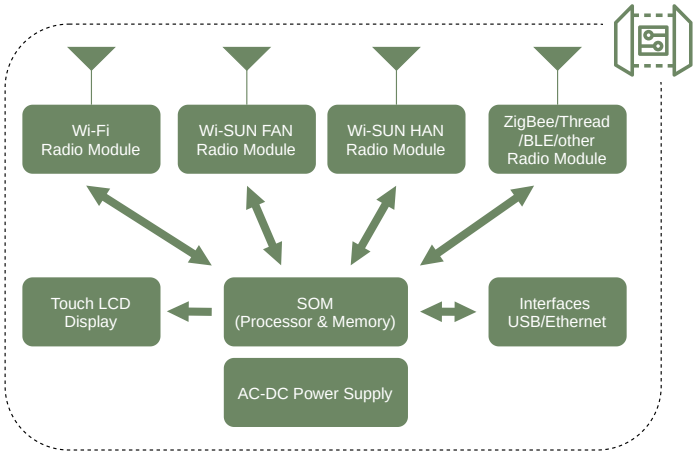


Figure 2. HEMS controller block diagram.

The hardware architecture is based on a flexible design approach using daughter boards for radio connections, such as Wi-SUN HAN, WiFi, other multiprotocol radios (ZigBee, Thread, BLE), and Wi-SUN FAN. The latter is used to connect the controller to the house's main smart meter, and can function an additional link to cloud services for low traffic scenarios.

Wi-SUN is a specification included in the IEEE 802.15.4 [29] standard, and contains both HAN and FAN profiles. The protocol was developed by Wi-SUN Alliance, particularly for use in utilities, smart cities, and IoT, defining the specifications of the PHY and MAC layers characterized as mesh, optimized, and secure networks. The supported encryption mode is Advanced Encryption Standard (AES), and authentication follows the Protocol for Carrying Authentication for Network Access (PANA). The applications of Wi-SUN include distribution automation, city lighting, smart parking, and environmental sensing [30].

ZigBee, Thread, and Wi-SUN are all based on the 802.15.4 protocol suite. These platforms compete with Bluetooth, though the latter has a limited range (10 m, or 10–50 m for BLE) [31]. Although widely used in many automation applications, WiFi has many drawbacks, especially regarding security, that make it not a good alternative for HEMS applications.

The choice to use Wi-SUN was based on the demand of the energy sector and smart cities for solutions adapted to these segments, which often require devices to transmit data

in several ways to provide redundancy in the transmission of information. In this context, the leading industrial-level protocols for IoT are Wi-SUN FAN and Wi-SUN HAN [1,2]. In addition to security, Wi-SUN provides lower latency, which is essential for applications where timing is a limiting factor. Devices supporting the FAN and HAN profiles are interoperable with other Wi-SUN certified devices.

It is important to mention that operation in the 900 MHz ISM band guarantees appropriate radio coverage even in buildings with masonry walls and houses with multiple floors. An outlet can relay messages from another outlet to the controller through a two-hop link, thereby extending the controller's coverage. In addition, the hardware architecture of the daughter boards facilitates new radio protocols and design improvements.

The processor and memory use a pluggable board for easy updates. The board uses an Advanced RISC Machine (ARM) processor to run Linux OS, the middleware, and local applications. The module form factor is System-on-Module (SOM) market standard, which has high-scale production and facilitates upgrades according to the demands of the software.

3.1.2. Smart Outlet

The smart outlets monitor the power consumption of the house's electrical appliances. Figure 3 shows the block diagram of a smart outlet.

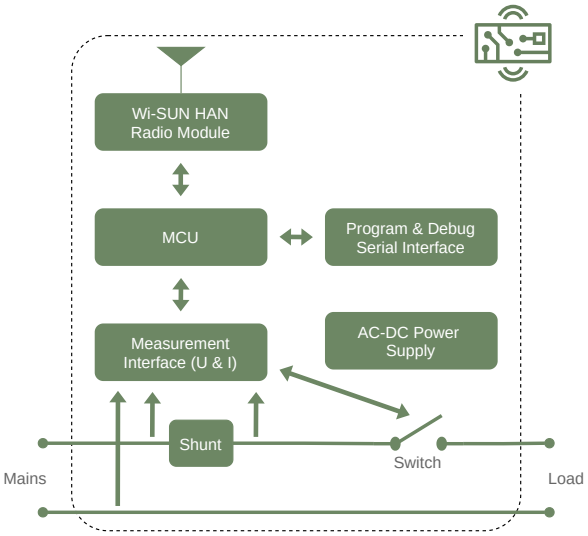


Figure 3. Smart outlet block diagram.

A single printed circuit board is equipped with a microcontroller unity (MCU) that measures the voltage and current of the load through a proper interface. The MCU calculates the active and reactive powers and periodically sends all these values (current, voltage, and power) to the HEMS controller. The smart outlet receives configuration parameters and commands to turn appliances on or off through middleware APIs. An internal protection switch disconnects the appliance in case of overcurrent, for which the value can be adjusted. The MCU firmware handles the radio transmissions and the initial pairing process to connect the outlet to the controller. The user can monitor the operating state through LEDs, use a push-button switch for pairing, and manually turn the load on or off. The outlet is simple to operate and easy to install in customers' premises. The design of the outlet allows for different encapsulation cases to fit varied installation scenarios and types of loads to be connected.

3.2. Middleware Architecture

The middleware provides an intermediary abstraction layer between all devices and the user applications (web and mobile apps). It facilitates secure data access and processing by providing high-level APIs, abstracting away the complexity and specificity of a physical

implementation. It provides a rich set of APIs for access from the cloud or mobile phones and locally to smart outlets or smart appliances. This software layer is essential to promote device interoperability as well as to address the HEMS market.

As shown in Figure 4, the middleware embedded in the HEMS controller is designed to work in low cost and low computational capacity hardware platforms.

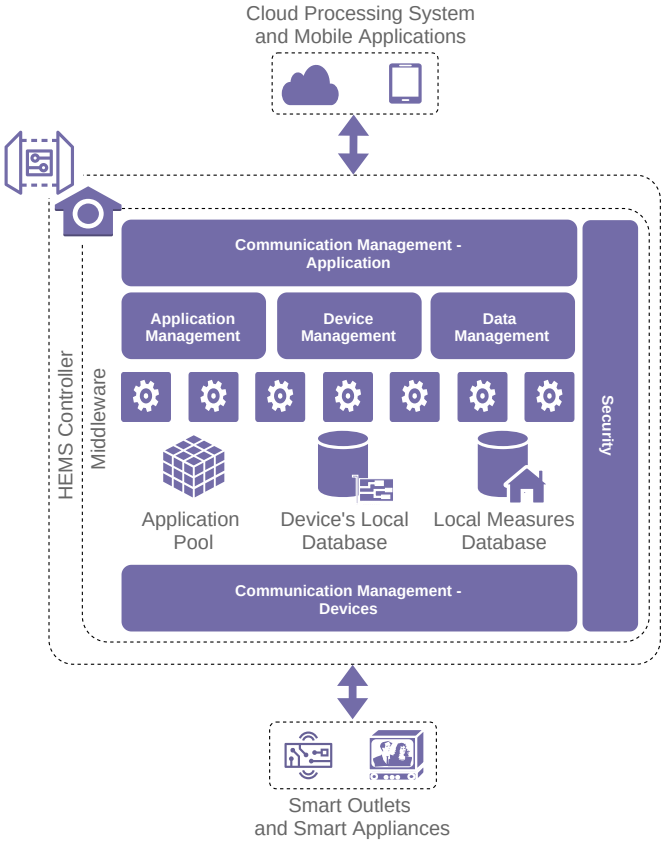


Figure 4. Middleware architecture.

It enables applications to be run closer to the end-user, consolidating an edge computing approach independent from external devices, e.g., mobile phones and PCs.

The middleware offers the following services: devices auto-detection, data acquisition, application acquisition, automatic updates, storage, control, and local processing. The following paragraphs present a brief overview of the proposed middleware architecture.

All functionalities are offered through an architecture based on microservices, providing higher resiliency and flexibility due to the independence of the services. Microservices provide a greater level of interoperability, as any device that meets the HTTP standard can communicate with the middleware through GET, POST, or PUT requests.

The middleware provides lower latency and eliminates dependence on external proprietary connections, improving the end-user experience through a local connection. The most complex functions may be delegated to the cloud, where more computing power is available. The functions offered by the cloud software are made available by the applications and websites.

The middleware architecture shown in Figure 4 consists of a core and a set of main modules. Each module is independent and has its own database. All features are accessed through the microservices. Python was the programming language used in developing the proposed middleware. Python provides a comprehensive set of tools for the implementation of modules. In addition, it has extensive cross-system portability and proven performance suitability for embedded systems. The latest techniques and architectures

for middleware development were utilized, including microservices, multiprocessing, and authenticated communication protocols.

The components of the architecture are described below:

- **Communication Management—Application:** the module responsible for remote communications, access control, sending data, and receiving messages. It has all the necessary functions to send the data collected and stored on cloud servers. A REST API was developed for communication with applications using a microservices architecture. Furthermore, the MQTT protocol was utilized to periodically send information to the cloud.
- **Application Management:** this module is responsible for managing applications running locally in the HEMS controller. These applications reside in the so-called application pool, where updating is possible through communication messages between the middleware and cloud processing system. This module was developed in Python utilizing the concept of multiprocessing, where we treated each application as an independent process.
- **Device Management:** the module responsible for managing local devices such as smart outlets and smart appliances. This module implements functions related to the devices, such as on/off actions, gathering current status, and controlling message sending. This module implements control APIs for the devices and, if necessary, interfaces with their proprietary APIs. Predefined APIs exist, and it is possible to develop and add new APIs.
- **Data management:** this module is responsible for managing the collected data. It is responsible for data acquisition and preparation as well as management and control of the local databases. We chose SQLite for the middleware database because it offers adequate performance for embedded systems and represents a well established solution. We implemented the database as a circular buffer, controlling its growth and defining its maximum size.
- **Communication management—Devices:** the module responsible for low-level communication with devices. This module has functions for reception, interface configuration, auto-detection, and registration of standards and device templates. This module features several possible device interfaces, which can be defined using the different templates available for various communication protocols. Furthermore, it includes APIs for protocols that utilize HTTP for communication, such as CoAP.
- **Security:** this an essential aspect of the architecture; it is introduced transversely, covering all modules and microservices. As no single technique is able to provide security to all the modules and microservices, a group of methods must be used. The architecture employs access control solutions, cryptography, authentication, and data integrity mechanisms.

As illustrated in Figure 4, the middleware relies on a local database to store the power measurements obtained by smart outlets before sending them to the cloud system. The databases have a circular buffer structure, allowing them to prioritize newer data. Furthermore, another database stores the smart outlets' current status and configuration parameters. The architecture includes middleware configuration microservices, which allows the middleware's behavior to be changed.

The middleware developed in this work is distinct from other middleware proposals found in the literature in that it uses an edge computing approach to bring the offered services closer to the end user. The objective behind edge computing is to reduce latency and dependency caused by the need for external communications, thereby increasing the performance of the offered services. The middleware proposal additionally stands out for using a microservices architecture in which it is possible to insert and change the offered services independently, thereby increasing the flexibility and scalability of the solution. To the best of our knowledge, an HEMS proposal with this feature has yet to be presented in the literature. It allows for large-scale applications while using inexpensive and computationally limited hardware.

The RT-DSP/SHArM Github project offers the open middleware implementation that enables the support of specifications for Home Energy Management System (HEMS) applications. The complete information regarding middleware can be found in [32].

3.3. Cloud Processing System

This system consists of microservices providing functionalities such as management of devices and users, data storage, and others. The cloud architecture diagram in Figure 5 depicts the description of the communication, storage, and interface stages:

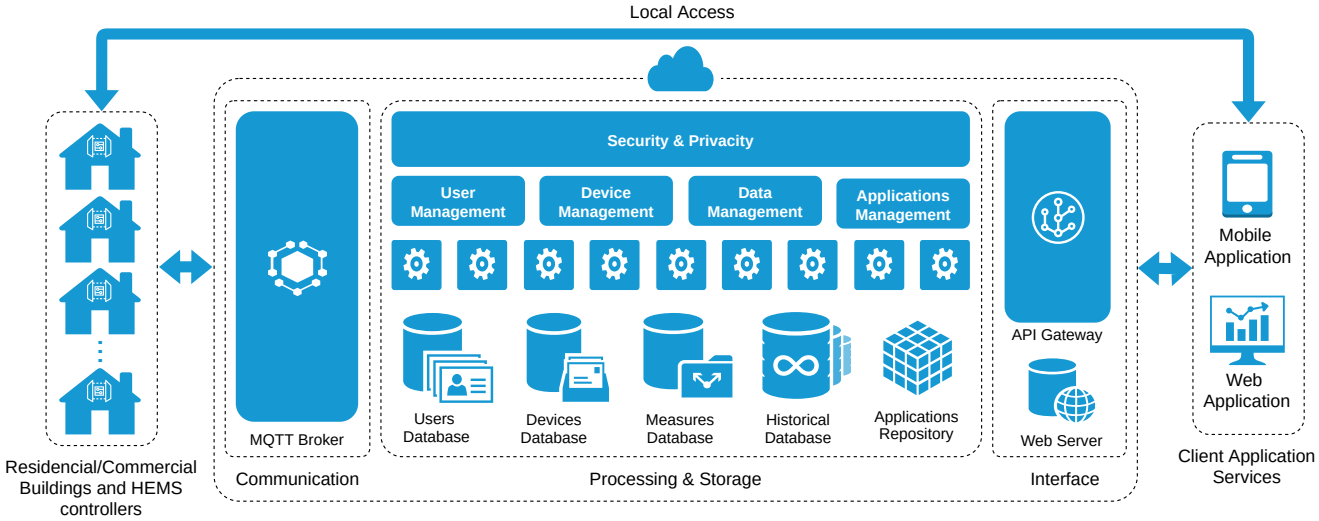


Figure 5. Cloud processing system architecture.

- **Communication Stage:** one requirement of the cloud system is that it needs to provide communication between the HEMS controllers and the cloud server. For this purpose, the system employs an MQTT broker running on the publish–subscribe model. The HEMS controllers publish the power measurements and device states in topics, then these data are sent for the processing stage. The MQTT broker operates as access control and authentication, providing the data to authenticated entities and enabling communication with authorized controllers only. It is possible to activate the use of the SSL/TLS protocol in addition to the QoS directives offered by the MQTT protocol.
- **Processing and Storage Stage:** the received data are validated and sent to their destinations. Each microservice performs a specific function, such as structuring, storing, and processing. The system is scalable, as additional microservices can be instantiated and used on demand.
  The defined databases are:
  - *User Database*—stores information regarding users and user–device relations.
  - *Device Database*—stores data from devices in general (i.e., HEMS controller and smart outlets), such as their status, configuration, description, and other information.
  - *Measures Database*—stores energy measurements in a structure, enabling quick time-based queries. It aims to improve the user experience by providing enhanced access to data in real time.
  - *Historical Database*—provides a central information repository containing overall and cumulative history, which can be used for analysis based on big data and other areas of data science.
- **Interface Stage:** the architecture features an API gateway, which is the single access point for client application services. The API gateway receives the requests and sends them to the requested service. Each service is assigned a specific communication protocol, such as REST AMQP, COaP, etc.

The proposed architecture is focused on data preparation and storage for new applications delivering to the edge. One improvement is the use of a microservices architecture to increase the overall scalability of the system, not only the scalability of the storage service [6–9]. Another advantage is the ease of updating and deploying new services.

The proposed system facilitates new data science-based applications using anonymous data for training AI models. For this purpose, a dedicated microservice prepares information while removing users' privacy information, and stores the anonymized data in a historical database.

Therefore, the cloud processing system offers all the resources needed for an HEMS application with automatic device detection, reception of new applications through cloud repositories, data reception through microservices independent of communication protocol, and customized configuration options.

We utilized various tools and techniques to develop the cloud processing system, such as MongoDB for general application management and InfluxDB for telemetry data. We employed JavaScript/TypeScript as the programming languages, with Node.js runtime, Express as the back-end API framework, and React employed for the front-end.

3.4. Integration of New Applications and Services

As shown in Figure 6, the architecture allows for easy deployment of new applications and services. The application repository is a crucial component in the cloud processing system that stores all the developed applications. For each deployment, there is a corresponding manifest message that describes the resources' name, type, and properties. Upon middleware restart or user demand, the exchange of these messages allows applications to be requested and downloaded. These processes keep applications updated in HEMS controllers or cloud pools, all of which contain managed applications.

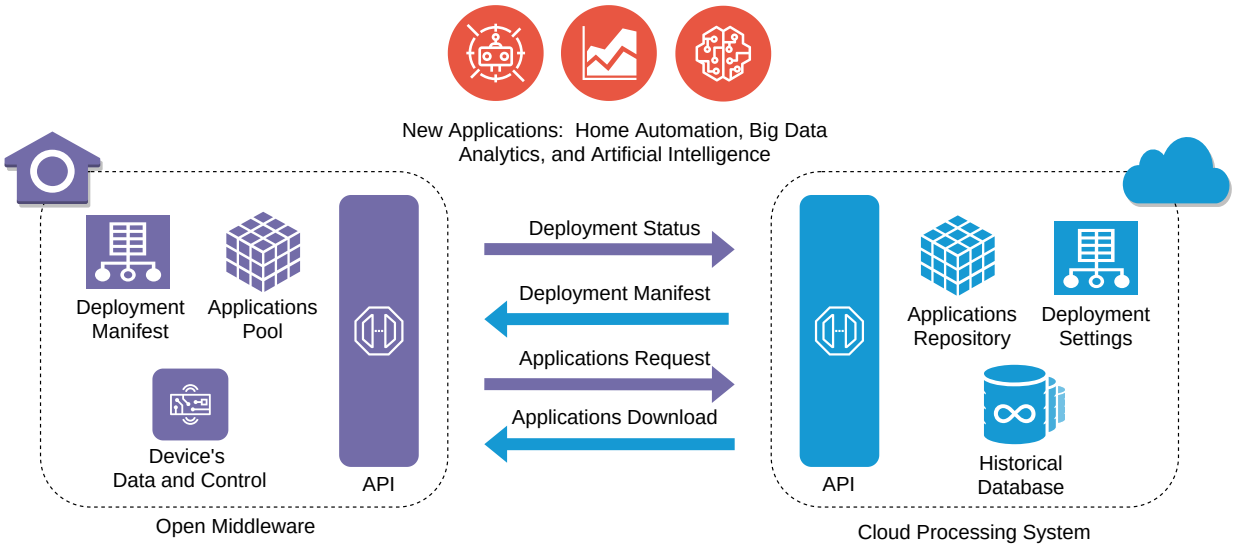


Figure 6. System architecture for the development and deployment of new applications.

4. Proof of Concept of the SHArM Architecture

This section presents the result of tests carried out to evaluate the overall system integrity and the load disaggregation model performance as a middleware application, as well as details of the client interfaces.

4.1. Flexible Hardware Design

After the smart outlets are energized, they are paired with the HEMS controller through an automatic/manual procedure. In this way, the middleware embedded in the controller can access the data and send commands to smart outlets. In addition, after the controller is connected to the internet via router, the middleware performs a communication

test with the cloud servers to later send data and receive commands. In case of cloud server connection failure, the middleware stores the data until it can send them later. These processes do not require user interventions.

Figure 7 shows the HEMS controller proof of concept, containing a main board according to the architecture depicted in Figure 2.



Figure 7. HEMS controller hardware photo.

Several wireless daughter boards are possible, such as ZigBee, Wi-SUN, WiFi, and Bluetooth. A display shows the controller status as well as customer energy consumption according to the selected smart outlet.

Figure 8 shows the smart outlet prototype, which has a female connector to plug in the appliance and to power the device in a very easy installation procedure at the customer's premises.

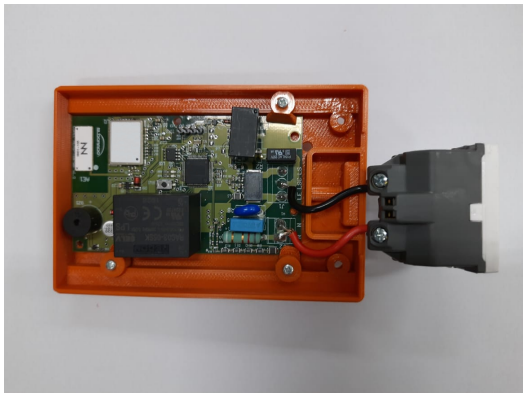


Figure 8. Smart outlet hardware photo.

It is important to note that the Wi-SUN HAN standard demonstrated significant robustness in wireless communication in the conducted field trials, particularly when compared to WiFi. This was observed even when using smart outlets behind stainless steel appliances such as refrigerators and dishwashing machines.

4.2. Middleware Application: Load Disaggregation

Load disaggregation is a non-intrusive load monitoring (NILM) process; it is used to individualize the power consumption of appliances by processing the aggregate measurements of power, current, and voltage for the entire premises [33]. Load disaggregation can be used to identity the on/off status of appliances as well.

As illustrated in Figure 9, in residential or commercial facilities where smart outlets are not available the load disaggregation method uses only the aggregated energy data from the smart meter.

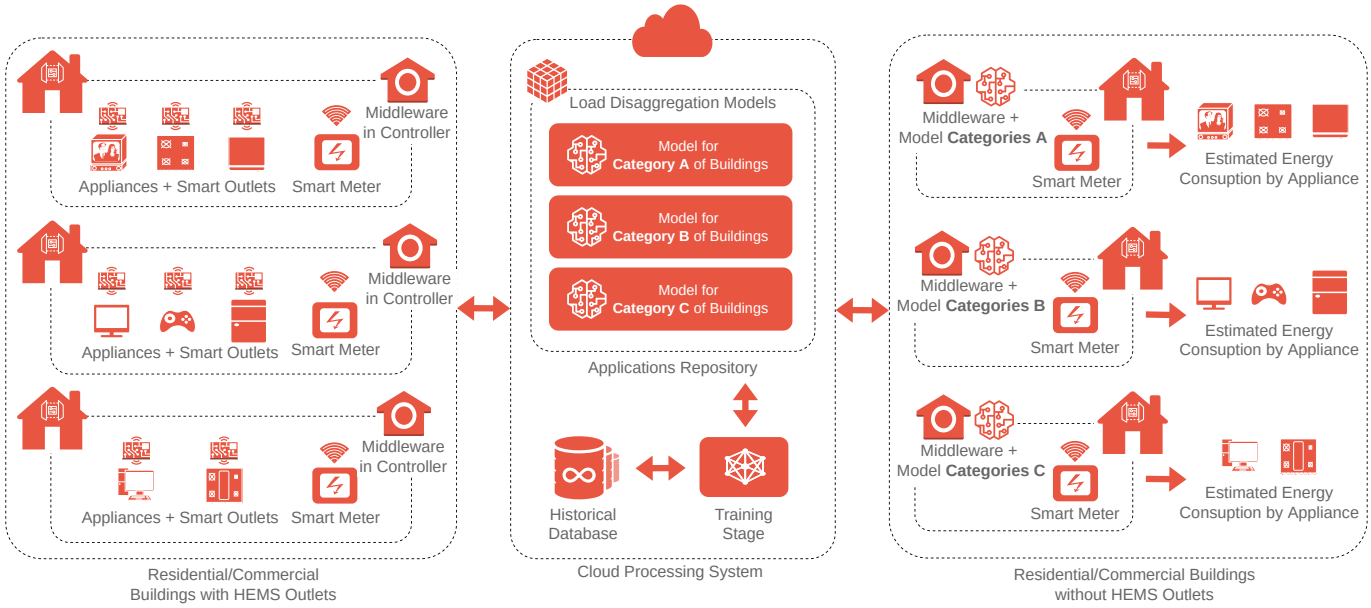


Figure 9. Load disaggregation block diagram.

The HEMS controller gathers the aggregated data collected by the smart meter via communication links employing the Wi-SUN HAN communication protocol. The training stage of the disaggregation method is performed in the cloud considering the data collected via smart outlets for individual loads and the aggregated data. The models resulting from the training stage define categories based on the set of appliances in the household. In this case, it enables the user to use the mobile application to select the model that most closely represents the residence. As a result, the disaggregation process run as a middleware application, providing the user with information regarding the appliances in operation and making it available for local or cloud accesses.

The disaggregation method used in this work is based on processing consumption measurements collected within a time window, as proposed in [3]. In this approach, the process consists of the two steps of load identification and disaggregation depicted in Figure 10. It is worth highlighting that this method considers only active power readings as features.

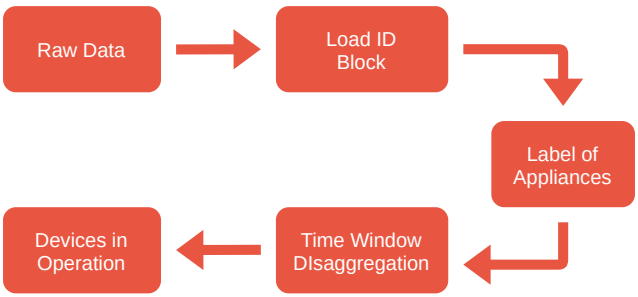


Figure 10. Flowchart of load disaggregation.

The load identification algorithm requests the raw data stored in the database related to the power readings of the smart outlets. We employed Principal Component Analysis (PCA) to identify the loads of disaggregation according to [3]. Consequently, we applied PCA in two stages. In the first stage, we used PCA to identify the loads; in the second stage, responsible for disaggregation, we applied PCA to extract the attributes. In the first stage, we organized the collected data from N appliances, d days, and s samples into an identification matrix $\mathbf{X}_{ID}$ [3]. Then, we applied PCA to the data matrix $\mathbf{X}_{ID}$. After applying this procedure, PCA produces the principal components. With the principal components, it is possible to generate clusters, with each cluster representing a household

appliance. In this step, the Accumulative Contribution Ratio (ACR) criteria sets the number of principal components considered in the problem. Thus, we formed clusters with the existing household appliances from the selected principal components, allowing us to generate labels for the existing appliances and thereby completing the process of load identification. At this point, the algorithm can initiate the disaggregation process. For the training phase, the first step is to establish the combinations of working devices according to the labels, determining the classes of the problem based on the on/off status of the appliances.

Next, the disaggregation process begins, using a time window that defines the interval corresponding to the vectors that compose the aggregated power data matrix. Subsequently, PCA is applied to reduce dimensionality, followed by feature extraction. A similar process is followed for the test data to send the patterns to the decision-maker's system. At this stage, a supervised learning algorithm identifies the operating devices. Machine learning algorithms with low computational complexity, such as k-NN, DT, and RF, can be used for this, as they are able to identify the appliances in operation while having very low computational complexity compared to deep learning models; hence, such machine learning algorithms are suitable for integration into our architecture. Further details regarding the time window algorithm are discussed in [3].

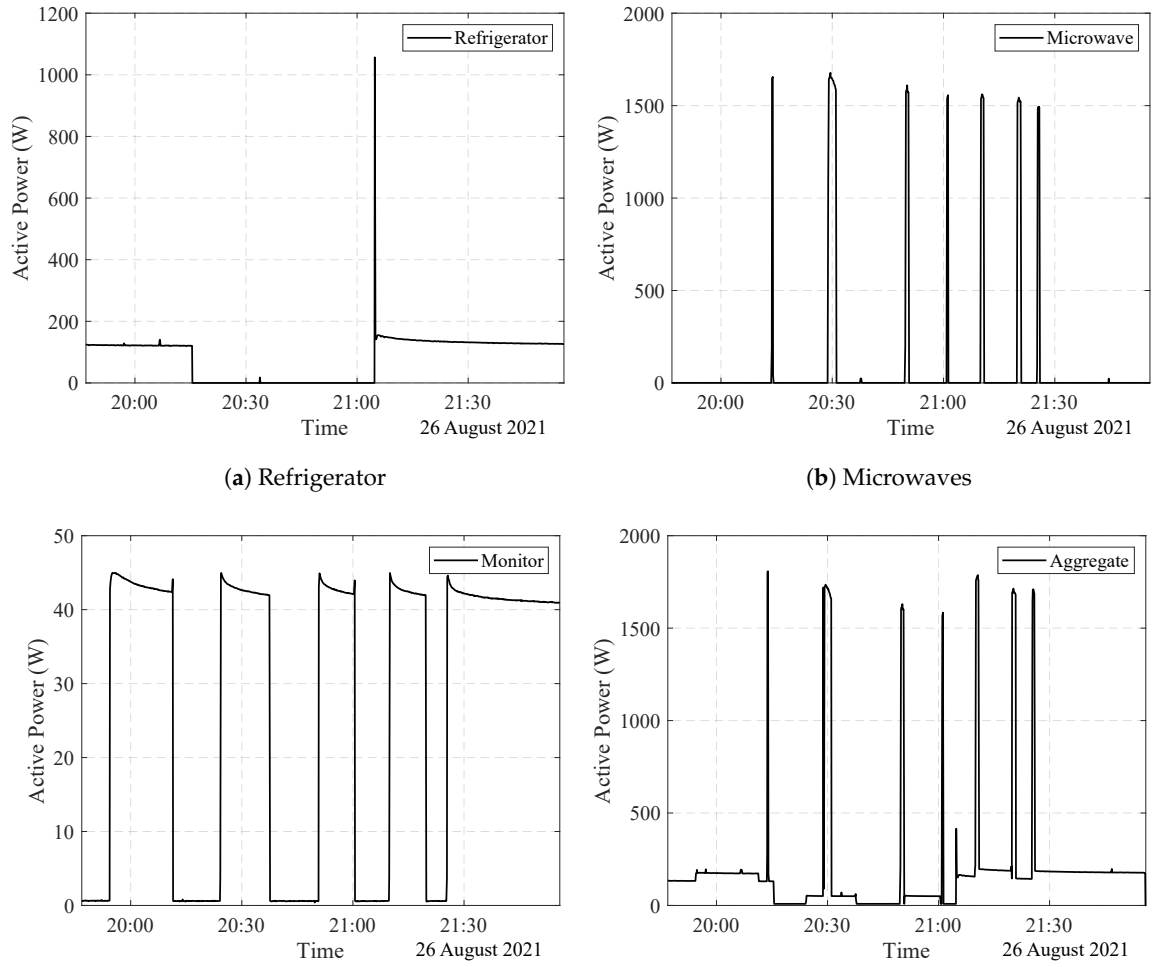
The algorithm's output consists of vectors of strings that denote the initial and final timestamps of the time window along with the list of appliances identified as being in the 'on' state.

Load Disaggregation Results

As previously mentioned the disaggregation method employed as part of the system works as an application of the middleware. After the benchmark tests, this approach was effectively tested using real data collected as part of this project. In the scenario described in Section III, the smart outlets gathered data from three household appliances: a refrigerator, a microwave, and a monitor. A fourth smart outlet collected data concerning the aggregated measurements of these three devices. It is necessary to mention that such smart outlets collected active power data in the described scenario. Furthermore, the smart outlets gathered data for 5 s each, i.e., at a measurement frequency of 1/5 Hz. From the data collection, we split the gathered data into two sets, one for training and other for testing. The split between the data used in the training and test phases was 60% and 40%, respectively. Figure 11 depicts the active power gathered from the previously mentioned devices and the aggregated data. After data collection, the middleware processes and sends the data to the cloud software for storage and database creation. The disaggregation algorithm requests the data from the cloud software and initiates the task according to the process described earlier in this section. The training stage is then executed to obtain the patterns required to perform the testing stage. The algorithm returns the list of devices identified as being in the 'on' state during the period corresponding to the time window; the results of the disaggregation process are shown in Figure 12. Figure 12 provides information regarding the timestamp of the readings performed by the outlets and the aggregate active power, where the colors represent the labels of combinations of appliances identified as being in the 'on' state at the investigated time intervals.

4.3. Client Application Services and Interfaces

To test the design of the client applications services, we focused on three possible users of HEMS: customers, specialists, and administrators. While all of these clients use the same web application, they have access to different services. The customer user application allows two possible connections: a local connection to the HEMS controller via WiFi network offline from the Internet, and a remote connection to the cloud server via the same REST API used by the web application.



(**a**) Refrigerator (**b**) Microwaves

(**c**) Monitor (**d**) Aggregated

Figure 11. (**a**–**c**): Active power of appliances connected to the smart outlets and (**d**) aggregated active power collected by the smart meter installed at the home energy input.

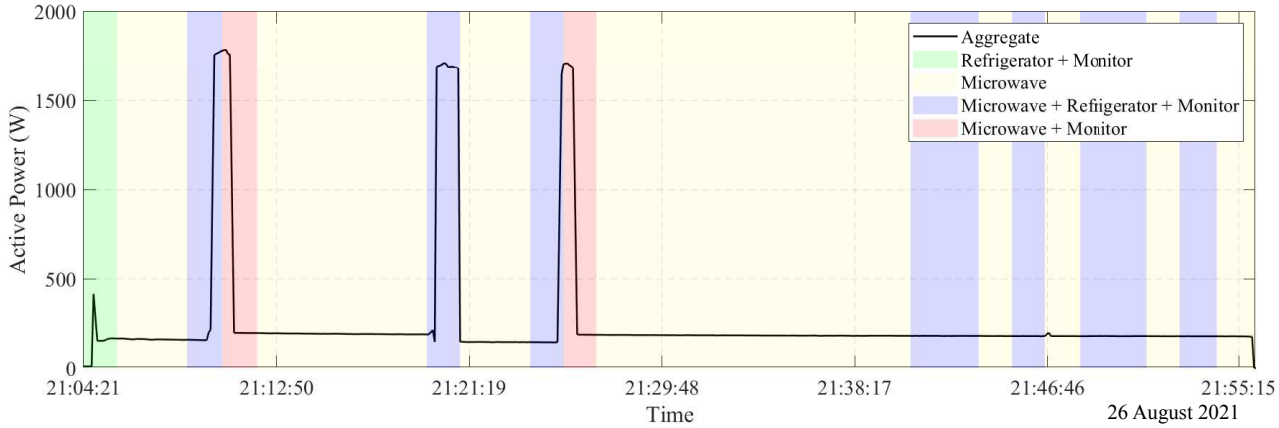


Figure 12. Aggregated active power values, highlighting the load disaggregation results.

The services available to each user profile are as follows:

- **Administrator Services**: for specialized professionals of the energy utility provider assigned as administrators, this user profile enables visualization of the current data

collection rate, storage capacity, and number of HEMS controllers, among other items. In addition, administrators are responsible for managing the system users and assigning the roles of specialists and new administrators.

- **Customer Services**: these services are dedicated to residential and commercial building owners. This user profile allows customers to visualize the energy consumption of each device along with the whole energy consumption. They can see the current device status and have an interface available to control their on/off status. The mobile application has a local connection that provides daily and weekly data as well as a remote connection that provides monthly and annually data.
- **Specialist Services**: these services are offered for specialists working at the energy utility provider, and involve managing tasks related to HEMS controllers installed in a given region. They allow users to aggregate the consumption measurements of a group of controllers. These services can be an important tool for energy providers. For instance, the analysis of these aggregated data can be useful for evaluating the demand of a region and identifying how it varies over time.

Interface Examples

The user interfaces can be used to manage smart outlets, associate each outlet with a given appliance, and indicate where in the house the outlet is located.

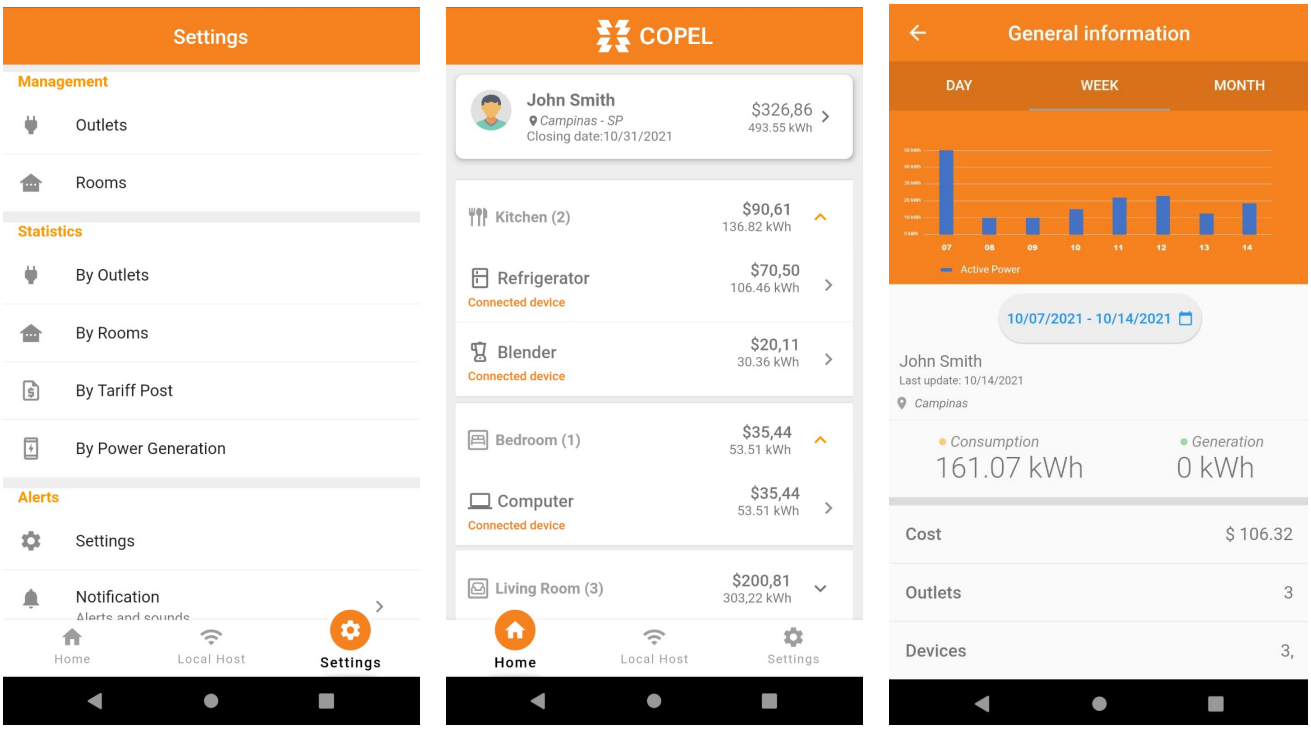
In the following, we present details of the client interfaces for the mobile application and web client pages.

Figure 13 presents three mobile application screens created for customer use. Figure 13a shows a portion of the options available in the application menu. The customers have two management options, namely, outlets and rooms. Statistics options allow users to view information generated by collecting data from the outlets while grouping outlets by room, energy consumption (according to the billing time unit), and power generation. Figure 13b shows the first menu option for the home, which presents crucial information about monetary costs to the customer. The screen displays each appliance's energy consumption (in kWh) and the aggregated power. On application startup, users have the option to select cloud or local service accesses. Several of the cloud services offered can be seen in Figure 13c, where the customer is shown a time series of energy consumption with daily, weekly, and yearly data. The local services (not shown here) allow the client to access data in a small historical interval and control their devices directly from the controller without the need for internet access.

Many of the features presented in the mobile application are available in the web application as well. Figure 14 shows an overview of the customers web application interface.

The main menu options include a dashboard that shows the main parameters of the system in a summarized format, including: management of the outlets, including their locations; statistics on the energy consumption patterns of smart outlets and appliances, outlet grouping, consumption by billing time unit, and power generation; alerts, including excessive energy consumption alerts available at the consumer's discretion; and system reports such as texts and graphics. Additionally, users can view existing user profiles and consult or change application settings.

Figure 15 shows an overview of the web application interface designed for the specialists and administrators of the energy provider. The interface presents additional technical measures which are not presented to the customer users. For instance, specialists and administrators can visualize power quality indicators based on nominal supplied voltage, power factor, and power outage, as well as values aggregated by regions. The main menu options include a dashboard (summarized information from the entire system), management (management of regions, controllers, outlets, users, and event notifications to be sent to users), system management (system information related to data communication and storage), and alerts (related to system problems and small power quality factor), among others functionalities.



(**a**) Menu Options. (**b**) Home screen. (**c**) Cloud Services.

Figure 13. Example of the mobile application interface.

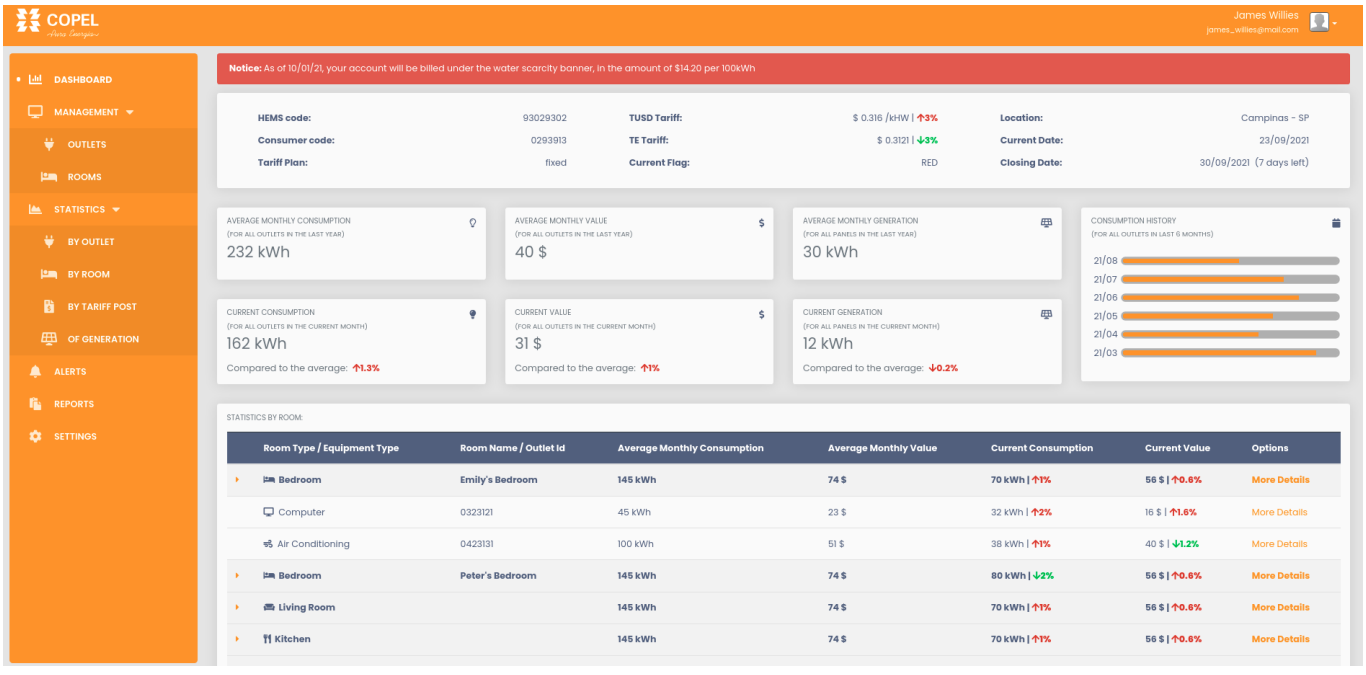


Figure 14. Customer dashboard screen of the web application.

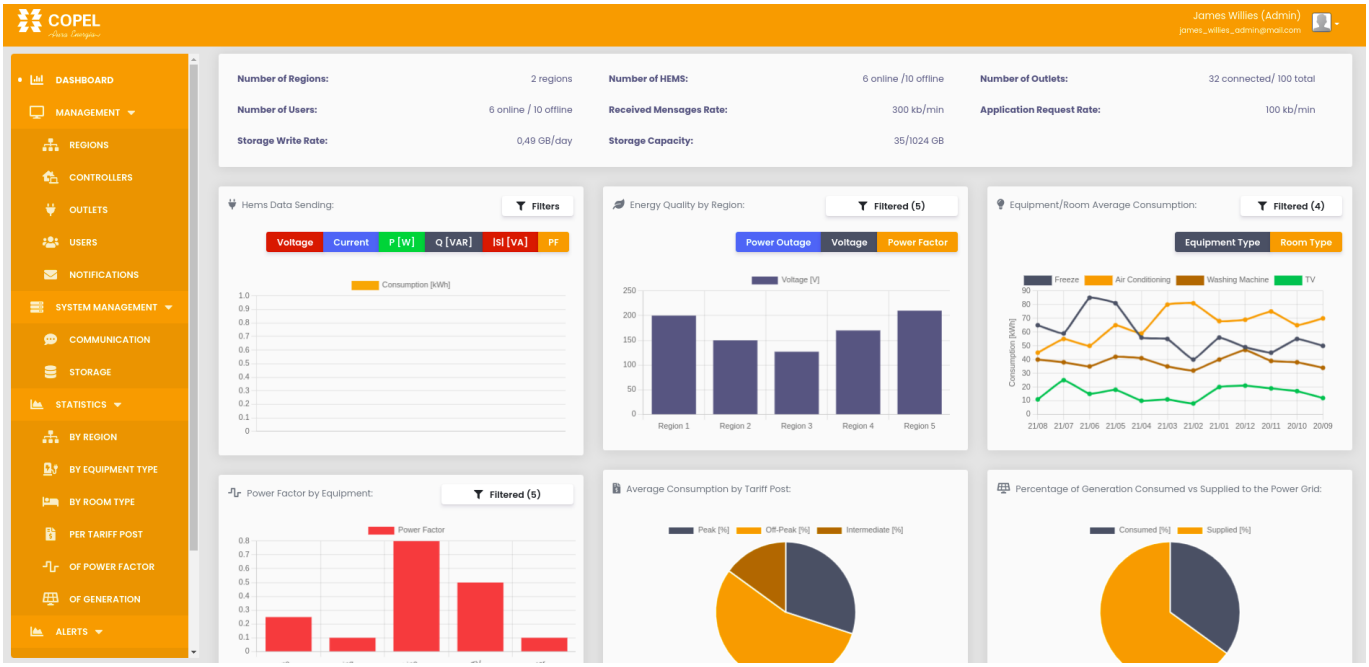


Figure 15. Specialist/administrator dashboard screen of the web application.

5. Conclusions

In this work, we have presented an innovative proposal comprising the main components of a Home Energy Management System (HEMS) based on a flexible architecture designed to facilitate hardware upgrades and enable the development of new edge computing applications. A flexible hardware architecture simplifies communication between smart outlets and appliances using different protocols beyond Wi-SUN HAN, such as Zigbee, Wi-Fi, BLE, Thread, etc.

An open middleware, available as a public Github project, offers a rich set of APIs for developing new applications at the edge and embedding intelligent services within the HEMS. Additionally, a cloud server is presented that can scale on demand and handle the vast amounts of data generated by smart meters and outlets. This server is equipped to store significant volumes of data, including historical consumption profiles, for use in future artificial intelligence and big data applications.

This work includes screenshots of value-added services for users in web and mobile applications desgined for both customers and energy supplier staff. Additionally, several relevant technical issues for HEMS manufacturers have been presented. Load disaggregation is incorporated within the system using the proposed APIs, which have been subjected to integrity tests.

Overall, the proposed architecture is expected to contribute significantly to the field of home energy management by inspiring new designs, enabling the implementation of policies and strategies for efficient energy use, and reducing the environmental impacts of energy generation.

Author Contributions: All authors contributed to the conceptualization, methodology, investigation, validation, and visualization of each part of this article. Software, L.L.M.; middleware, L.C.B.C.F., G.d.S.C. and A.B.; artificial intelligence, T.W.C. and D.A.M.L.; writing—original draft preparation, L.L.M. and L.C.B.C.F.; writing—review and editing, P.C., G.F., and L.G.P.M.; supervision, F.B.N.; project administration, E.R.d.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Companhia Paranaense de Energia under Grant COPEL Projet: ANEEL-PD-02866-0508/2019.

Data Availability Statement: The access to the data underlying the findings of this study is not available due to privacy considerations and in accordance with company operational policies.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACR	Accumulative Contribution Ratio
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARM	Advanced RISC Machine
BLE	Bluetooth Low-Energy
COaP	Constrained Application Protocol
DSM	Demand-Side Management
HEMS	Home Energy Management Systems
HTTP	Hypertext Transfer Protocol
FAN	Field Area Network
HAN	Home Area Network
IoT	Internet of Things
LCD	Liquid Crystal Display
MAC	Media Access Control
ML	Machine Learning
MCU	Microcontroller Unity
MQTT	Message Queuing Telemetry Transport
NILM	Non-Intrusive Load Monitoring
OS	Operational System
PANA	Carrying Authentication for Network Access
PCA	Principal Component Analysis
PHY	Physical Layer
PLC	Power Line Communication
PoC	Proof of Concept
QoS	Quality of Service
REST	Representational State Transfer
SHArM	Smart HEMS Architecture and Middleware
SG	Smart Grid
SOM	System-on-Module
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TLS	Transport Layer Security
UI/UX	User Interface/User Experience Design
Wi-SUN	Wireless Smart Ubiquitous Network
WSN	Wireless Sensor Networks

References

1. Harada, H.; Mizutani, K.; Fujiwara, J.; Mochizuki, K.; Obata, K.; Okumura, R. IEEE 802.15. 4g based Wi-SUN communication systems. *IEICE Trans. Commun.* **2017**, *100*, 1032–1043. [CrossRef]
2. Mizutani, K.; Okumura, R.; Mizutani, K.; Harada, H. A Home Area Heterogeneous Wireless Management Scheme by Wi-SUN FAN and Wi-Fi Systems. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–4. [CrossRef]
3. Lemes, D.A.M.; Cabral, W.T.; Fraidenraich, G.; Meloni, L.G.P.; De Lima, E.R.; Neto, F.B. Load Disaggregation Based on Time Window for HEMS Application. *IEEE Access* **2021**, *9*, 70746–70757. [CrossRef]
4. Han, J.; Choi, C.S.; Lee, I. More efficient home energy management system based on ZigBee communication and infrared remote controls. *IEEE Trans. Consum. Electron.* **2011**, *57*, 85–89.
5. Han, J.; Choi, C.S.; Park, W.K.; Lee, I.; Kim, S.H. Smart home energy management system including renewable energy based on ZigBee and PLC. *IEEE Trans. Consum. Electron.* **2014**, *60*, 198–202. [CrossRef]

6. Al-Ali, A.R.; Zualkernan, I.A.; Rashid, M.; Gupta, R.; Alikarar, M. A smart home energy management system using IoT and big data analytics approach. *IEEE Trans. Consum. Electron.* **2017**, *63*, 426–434. [CrossRef]
7. Zhou, S.; Wu, Z.; Li, J.; Zhang, X.P. Real-time energy control approach for smart home energy management system. *Electr. Power Components Syst.* **2014**, *42*, 315–326. [CrossRef]
8. Shareef, H.; Al-Hassan, E.; Sirjani, R. Wireless home energy management system with smart rule-based controller. *Appl. Sci.* **2020**, *10*, 4533. [CrossRef]
9. Pawar, P.; TarunKumar, M.; Panduranga, V.K. An IoT based Intelligent Smart Energy Management System with accurate forecasting and load strategy for renewable generation. *Measurement* **2020**, *152*, 107187. [CrossRef]
10. Aman, S.; Simmhan, Y.; Prasanna, V. Energy Management System: State of Art and Emerging Trends. *IEEE Commun. Mag.* **2013**, *51*, 114–119. [CrossRef]
11. Liu, Y.; Qiu, B.; Fan, X.; Zhu, H.; Han, B. Review of smart home energy management systems. *Energy Procedia* **2016**, *104*, 504–508. [CrossRef]
12. Shareef, H.; Ahmed, M.S.; Mohamed, A.; Al Hassan, E. Review on home energy management system considering demand responses, smart technologies, and intelligent controllers. *IEEE Access* **2018**, *6*, 24498–24509. [CrossRef]
13. Mahapatra, B.; Nayyar, A. Home Energy Management System (HEMS): Concept, architecture, infrastructure, challenges and energy management schemes. *Energy Syst.* **2022**, *13*, 643–669. [CrossRef]
14. Sharda, S.; Singh, M.; Sharma, K. Demand side management through load shifting in IoT based HEMS: Overview, challenges and opportunities. *Sustain. Cities Soc.* **2021**, *65*, 102517. [CrossRef]
15. Edwards, W.; Barron, F.H. SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. *Organ. Behav. Hum. Decis. Process.* **1994**, *60*, 306–325. [CrossRef]
16. Capehart, B.L.; Muth, E.J.; Storin, M.O. Minimizing Residential Electrical Energy Costs Using Microcomputer Energy Management Systems. *Comput. Ind. Eng.* **1982**, *6*, 261–269. [CrossRef]
17. Prince, B.; Sasson, A.; Wynne, W.; Trefny, F.; Cleveland, F.; Robinson, J.T.; Horton, J. Advances in energy management systems. *IEEE Trans. Power Syst.* **1986**, *1*, 226–234. [CrossRef]
18. Hopley, I.; Holden, T.; Wilhelmij, G. Decision support for energy management systems. In Proceedings of the International Conference on Information-Decision-Action Systems in Complex Organisations, Online, 6–8 April 1992; pp. 95–99.
19. Kushiro, N.; Suzuki, S.; Nakata, M.; Takahara, H.; Inoue, M. Integrated residential gateway controller for home energy management system. *IEEE Trans. Consum. Electron.* **2003**, *49*, 629–636. [CrossRef]
20. Inoue, M.; Higuma, T.; Ito, Y.; Kushiro, N.; Kubota, H. Network architecture for home energy management system. *IEEE Trans. Consum. Electron.* **2003**, *49*, 606–613. [CrossRef]
21. Han, B.; Zahraoui, Y.; Mubin, M.; Mekhilef, S.; Seyedmahmoudian, M.; Stojcevski, A. Home Energy Management Systems: A Review of the Concept, Architecture, and Scheduling Strategies. *IEEE Access* **2023**, *11*, 19999–20025. [CrossRef]
22. da Cruz, M.A.; Rodrigues, J.J.P.; Al-Muhtadi, J.; Korotaev, V.V.; de Albuquerque, V.H.C. A reference model for internet of things middleware. *IEEE Internet Things J.* **2018**, *5*, 871–883. [CrossRef]
23. Nagalakshmi, S.; Prabha, M.; Senthamarai, R.; Rohini, G. Design and implementation of Arduino based smart home energy management system using renewable energy resources. *Int. J. Chemtech Res.* **2017**, *10*, 696–701.
24. Oliveira, E.d.L.; Alfaia, R.D.; Souto, A.V.; Silva, M.S.; Francês, C.R.L.; Vijaykumar, N. Smartcom: Smart consumption management architecture for providing a user-friendly smart home based on metering and computational intelligence. *J. Microwaves Optoelectron. Electromagn. Appl.* **2017**, *16*, 736–755. [CrossRef]
25. Rodríguez-Molina, J.; Kammen, D.M. Middleware architectures for the smart grid: A survey on the state-of-the-art, taxonomy and main open issues. *IEEE Commun. Surv. Tutorials* **2018**, *20*, 2992–3033. [CrossRef]
26. Machorro-Cano, I.; Alor-Hernández, G.; Paredes-Valverde, M.A.; Rodríguez-Mazahua, L.; Sánchez-Cervantes, J.L.; Olmedo-Aguirre, J.O. HEMS-IoT: A big data and machine learning-based smart home system for energy saving. *Energies* **2020**, *13*, 1097. [CrossRef]
27. Reyes-Campos, J.; Alor-Hernández, G.; Machorro-Cano, I.; Olmedo-Aguirre, J.O.; Sánchez-Cervantes, J.L.; Rodríguez-Mazahua, L. Discovery of resident behavior patterns using machine learning techniques and IoT paradigm. *Mathematics* **2021**, *9*, 219. [CrossRef]
28. Taiwo, O.; Ezugwu, A.E. Internet of Things-Based Intelligent Smart Home Control System. *Secur. Commun. Netw.* **2021**, *2021*, 9928254. [CrossRef]
29. *IEEE Std 802.15.4-2003*; Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE: Piscataway, NJ, USA, 2003.
30. Wi-SUN Alliance. Wi-SUN Alliance: Promoting Interoperable Internet of Things Standards for Smart Utilities and Cities. 2011. Avaiable online: https://www.wi-sun.org/ (accessed on 20 May 2022).
31. Dobrilović, D. Networking technologies for smart cities: An overview. *Interdiscip. Descr. Complex Syst. INDECS* **2018**, *16*, 408–416. [CrossRef]

32. Ferreira, L.C.B.C.; Borchardt, A.D.R.; Cardoso, G.D.S.; Mendes Lemes, D.A.; Sousa, G.R.D.R.d.; Neto, F.B.; de Lima, E.R.; Fraidenraich, G.; Cardieri, P.; Meloni, L.G.P. Edge Computing and Microservices Middleware for Home Energy Management Systems. *IEEE Access* **2022**, *10*, 109663–109676. [CrossRef]
33. Ruano, A.; Hernandez, A.; Ureña, J.; Ruano, M.; García, J. NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review. *Energies* **2019**, *12*, 2203. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.