Scientific Research Publishing

Journal of Computer and Communications, 2023, 11, 64-81
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Smart Home Energy Monitoring System Based on Internet of Things and Inter Planetary File System for Secure Data Sharing

Aytun Onay[1,2*], Gökhan Ertürk[3], Cem Kıranlı[1], Hande Ateş[3], Yunus E. Isikdemir[1]

[1]Department of Research and Development, Inovasyon Muhendislik Ltd. Sti., Osmangazi Teknopark, Eskisehir, Türkiye
[2]Department of Software Engineering,Turkish Aeronautical Association University, Ankara, Türkiye
[3]ACD Bilgi İşlem Bilgisayar Yazılım Hizmetleri San ve Tic. Ltd. Sti., Osmangazi Teknopark, Eskisehir, Türkiye
Email: *aonay@thk.edu.tr

How to cite this paper: Onay, A., Ertürk, G., Kıranlı, C., Ateş, H. and Isikdemir, Y.E. (2023) A Smart Home Energy Monitoring System Based on Internet of Things and Inter Planetary File System for Secure Data Sharing. *Journal of Computer and Communications*, **11**, 64-81.
https://doi.org/10.4236/jcc.2023.1110005

Received: August 30, 2023
Accepted: October 21, 2023
Published: October 24, 2023

Abstract

Energy demand will continue to rise as a result of predicted population growth. In this work, a user-friendly home energy monitoring system based on IoT is described, which is capable of collecting, analyzing, and displaying data. Users register their sensors and devices on the monitoring platform. PostgreSQL and Elasticsearch databases are used to store the resulting measurements. In a smart home, the wireless sensor ACS712 was used to monitor the flow of electricity (current and voltage) for a household device. The user can share data about electricity consumption and costs with a third party via the private IPFS (InterPlanetary File System) network. A third party can download all the energy consumption data for a device or many devices from the platform for 1 day, 3 months, 6 months, and 1 year. The studies on the development of energy-efficient technology for home devices benefit greatly from the gathered data. For security in the system, it is preferred to run Keyrock Idm, Wilma Pep Proxy, and Orion Context Broker in HTTPS mode, and MQTTS is used to retrieve sensor data. The experimental results showed that the energy monitoring system accurately records voltage, current, active power, and the total amount of power used and offers low-cost solutions to the users using household devices in a day.

Keywords

Energy Monitoring System, MQTT, Fiware Architecture, The ACS712 Wireless Sensor, Smart Home

1. Introduction

Many nations around the world are becoming more interested in and giving

importance to the development of energy efficiency and renewable energy technology [1] [2] [3] [4]. Two of the main objectives of European policies in the area of research and innovation are energy conservation and the advancement of information and communication technologies (ICT) [5]. These goals aim to both accelerate the adoption of cutting-edge technological solutions and to mitigate climate change by lowering CO_2 emissions [6] [7]. According to Eurostat, as much as 39.2% of Europe's 2016 final energy consumption came from the residential sector, which accounted for 25.7% of total consumption. This puts buildings firmly in the spotlight when it comes to energy use [8]. Certain steps must be taken in order to reduce the consumption of electrical energy. The amount of electricity used by an appliance is measured in order to determine how it is being used. It is easier to reduce a home's electricity consumption if this practice is learned. The Internet of Things (IoT) is a network of physical items that are integrated with specific types of electronics, allowing them to connect and exchange data with one another [9] [10] [11] [12]. Technologies from IoT have played a significant role in managing energy utilization, enabling residents to better manage their resources [13]. As a result, electricity theft is also prevented. Sensors are used to track consumption in real time [14]. The InterPlanetary File System (IPFS) network is used by our monitoring platform to share data in a private way. IPFS is a peer-to-peer distributed file system that connects all computers to the same file system. IPFS is like the Web, but it's a BitTorrent swarm. IPFS supports content-addressed block storage with hyper connections [15] [16]. The study demonstrates the automation of the energy meter used to assess the energy spent by appliances and the total energy consumption over a period of days, months, and years. The following are the main contributions of this paper:

- An IoT-based energy monitoring system was designed to track the energy and power use of home appliances in real time.
- In our platform, users register their sensors and devices, and then PostgreSQL and Elasticsearch databases are used to store the resulting measurements. The ACS712 wireless sensor is used to measure the flow of electricity to a household device.
- The user can share the data on electricity consumption and costs with a third party using the private IPFS network. This information is gathered from household devices that are registered to the monitoring system.

The rest of the paper is structured as follows. Relevant studies on smart home energy monitoring and/or management systems are presented in Section 2. In Section 3, a design of an IoT energy monitoring system is presented to monitor the energy and power consumption of household devices in real time. Several experimental results of the energy monitoring platform are given in Section 4. Section 5 details the conclusion and future works.

2. Related Work

A high quality of life is currently being influenced by smart home networks. IoT

will soon be able to connect even the most basic household appliances to the internet. In the near future, a smart home will be able to work well because it will have a standard infrastructure that allows multiple wireless protocols to work together [17] [18] [19] [20]. Fletcher *et al.* have developed a user-friendly, low-cost home energy monitoring and recording system. The system allows users to monitor power usage and remotely control and replace their electronic devices using any web-enabled device. The system collects information about how people use energy and shows this information to make people more aware of how much energy they use [21]. The system can be improved in terms of data security and sharing. In our monitoring platform, from sensors to the Orion Context Broker, all data transmission will be secured. MQTT also supports user and password authentication. To prevent a user from listening to another user's topics, Access Control Lists (ACLs) are created in MQTT. Users can only listen to their own topics and cannot publish or subscribe to topics created by other users. For security in the system, it is preferred to run Keyrock Idm, Wilma Pep Proxy, and Orion Context Broker in HTTPS mode, and MQTTS is used to retrieve sensor data. Tran *et al.* provide a multi-dimensional hybrid strategy that combines multi-interactions between simulation-based and observation-based data using energy modeling's graphical interface software. Energy modeling and monitoring simplifies the estimation of energy use with the addition of residential occupancy and other key residential information and informs residents with energy performance reports [22]. Al-Kuwari *et al.* designed an IoT-based sensing and monitoring system for smart home automation, which uses the EmonCMS platform for collecting and visualizing monitored data and remote controlling of home appliances. The sensing of different variables inside the house is conducted using the NodeMCU-ESP8266 microcontroller board. The Smart Solar House's planned design looks to be flexible. It is said that it is easy to grow by adding more sensors, controllers, and measurement parameters, and that it can be used in bigger buildings [23]. In another study, Srinivasan *et al.* established a smart and sustainable technology based on IoT, utilizing Smart Plugs for data capture, which is then communicated through the wireless-gateway to the central database. Different strategies are designed to recognize consumer behavior and take the appropriate actions to reduce overall energy consumption using the data collected from the smart plug [24].

The monitoring platform developed in this study uses a private IPFS network for data sharing. The system should store data in IPFS and securely share data using blockchain technology. IPFS will provide a publicly accessible database, while blockchain makes it publicly verifiable. So it would be much better to take the best parts of each protocol and use them. Also, their solutions take the most basic security requirements into account (confidentiality, integrity, privacy, and access control). From this point of view, blockchain technology should be added to our system to make it more secure. Multiple factors guarantee the security of the blockchain: cryptography, consensus algorithms, immutability, traceability,

and the replication of data across all nodes.

3. Monitoring System Architecture

The monitoring system collects several sets of data from distinct sources in order to provide software applications to automatically compute and present a set of KPIs. The proposed system was designed to consist of four components: a device layer (end device), a communication layer (the gateway), a data layer (application server and databases), and an information layer (dashboard and analysis). The Bluetooth 5 protocol has been used to establish a connection between the end-devices and the gateway. The 6LowPAN protocol has also been performed for communication [25] [26]. The IPv6 over Low Power Wireless Personal Networks protocol works in accordance with IEEE's 802.15.4 standards and provides multiple devices to communicate with devices (sensors, triggers, etc.) in the physical world and also uses lots of different data types with less energy. Solutions for Smart Homes have been improved by using Fiware, which performs a universal set of standards for context data management. The Fiware Context Broker is the core component of our platform, making it possible for the system to perform updates and access the current state of context. IoT platforms with wireless communication have become more popular in recent years for fast and efficient data transfer and analysis. Thus, smart systems are of great importance for IoT technologies. In this study, the monitoring system was developed in order to build a system for sharing, storing, visualizing, and analyzing data. The system was broken up into three parts: data transfer (fiware), the user interface (front end), and the backend. In the next sections, these groups are discussed in detail.

3.1. Fiware Architecture

Fiware is architecture for creating communication systems between sensors and databases [27]. Fiware architecture was presented in compact form for simplicity in the block diagram in **Figure 1**. This architecture can be divided into 3 subgroups containing data storing, security, and data sharing. These groups are explained in detail in the following sections.

3.1.1. Data Storing and Data Flow

In our platform, users register their sensors and devices, and the measurements coming from these sensors are stored in PostgreSQL and Elasticsearch databases. Platform adapters, IoT agents, PEP Proxy, Orion Context Broker, MongoDB, Draco, and PostgreSQL components are included in the data storage section [28] [29] [30] [31].

Platform adapters are the communication protocols that are responsible for transferring data from sensors to the database. The adapters consist of HTTP, MQTT, or other protocols. The MQTT protocol is used for transferring data. IoT agents are responsible for users registering their sensors and devices. Devices are registered in the IoT Agent. The user provisions a sensor group of a device

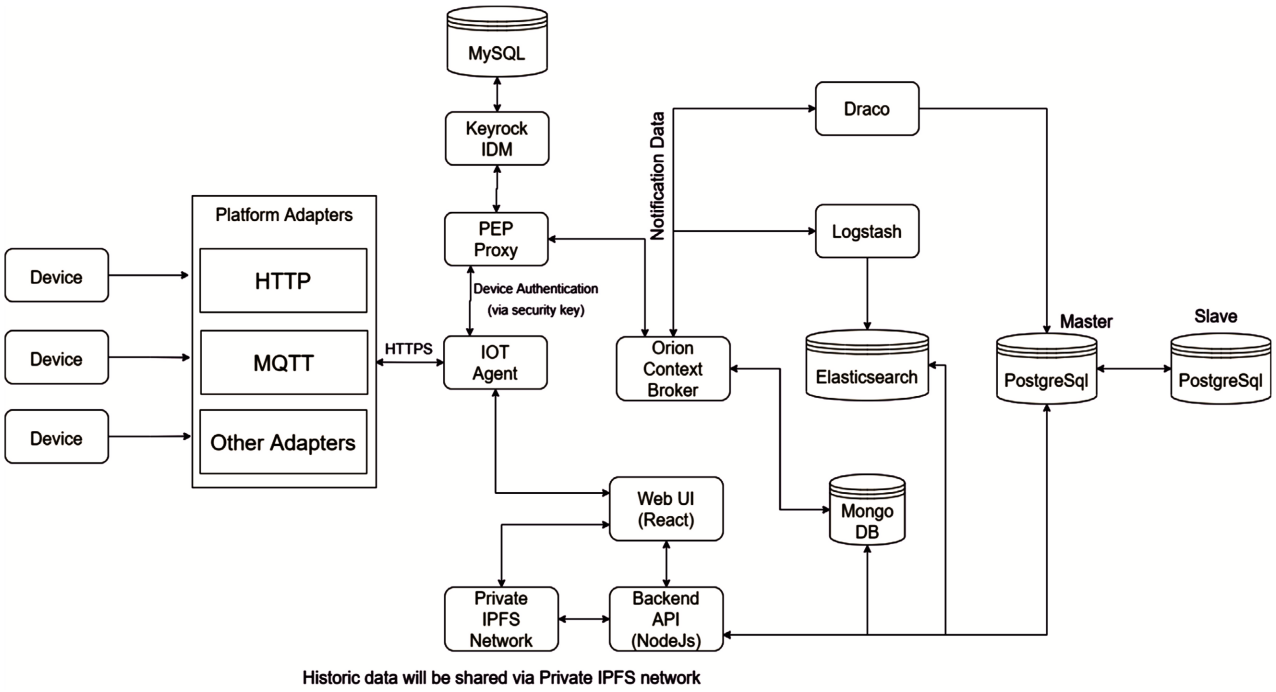

Figure 1. Fiware architecture.

for a specific fiware-service and fiware-service path and assigns a security key to this sensor group to be used for authentication. Then, the user defines the devices under the specific group by giving them sensor ids. The security key for the group and the sensor ids have been used for the flow of the sensor data in the system. From that point on, the measurement data from the specific sensor group is published to an MQTT topic in "securitykey/sensorid/attrs" form. Since device registrations are performed here, the IoT Agent knows what this format indicates and is able to subscribe to and listen for traffic coming from the sensors. It also maps the data coming from this channel to related attributes in the context broker. This way provides a secure data sending channel and serves as a private way of sending sensor data. Therefore, the security key should be kept secret by the user. In the process of provisioning a sensor group, the user's login credentials and a trust token from Keyrock are also added to the sensor group, which are then used to identify the sensor group when communicating with Wilma PEP Proxy. This trust token will be added as a header to all sensor data traffic, and sensor authentication will take place in the Wilma Pep Proxy before reaching the Orion Context Broker. Wilma Pep Proxy is combined with Keyrock to provide access control to Orion Context Broker. In the event of this trust token not being valid, sensor measurement data will not be accepted. Orion and IoT agents use MongoDB databases to hold persistent information they need to operate. As a result of the subscription of Draco and Logstash to Orion Context Broker, Draco and Logstash will be informed of all the context data changes, and the persisting historic context data will be stored in the PostgreSQL and Elasticsearch databases. Orion Context Broker is responsible for managing information

in terms of updates, queries, registrations, and subscriptions. As shown in Figure 2, the NGSI data model is made up of three main parts: context entities, attributes, and metadata.

MongoDB is an open source NoSQL document-store database, which is designed to handle JSON documents. In the Fiware architecture, MongoDB holds the entity and subscription for data coming into the Orion context broker. To store the data, Draco was used in this study. Draco is a simple-to-use, powerful and reliable data processing and distribution system. Internally, Draco is built on Apache NiFi, a data flow system based on flow-based programming ideas. It offers directed graphs of data routing, transformation, and system mediation logic that are both powerful and scalable. It was created to automate data transfer across systems. The integrated data logistics platform, Apache NiFi, was presented in Figure 3. In this content, the word "dataflow" is used to refer to the automated and regulated flow of data across systems.

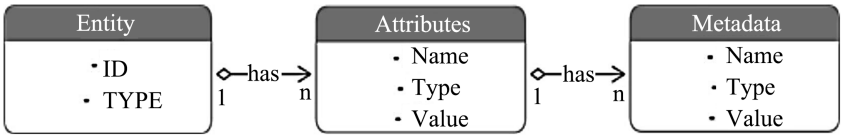

Figure 2. Orion context broker.

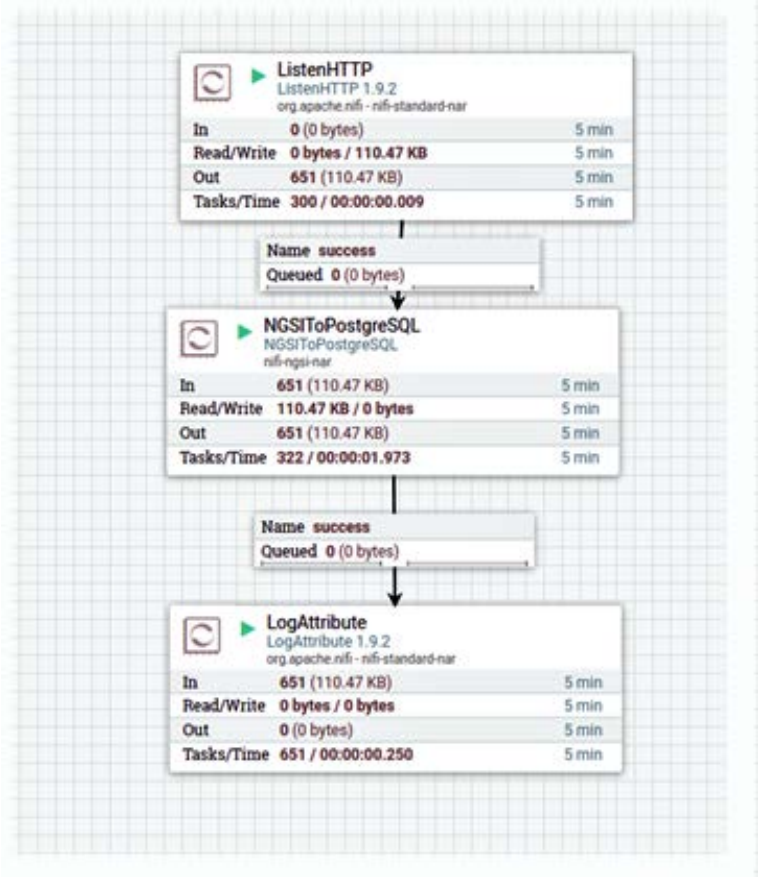

Figure 3. Apache NiFi screen.

PostgreSQL is an open source relational database system, which is also called a SQL database. In this study, the main purpose of this database is to save the specified fiware-service and fiware-service path from the data that comes to Orion Context Broker.

3.1.2. Data Sharing

In our platform, data sharing is done via a private IPFS network. A third party can download data from the platform, which has organizations that allow the data to be shared. Then, the third parties select the domain and subdomain of the organization and specify the time range of the data. You have four different options to choose from: 1 month, 3 months, 6 months, and 1 year. For example, if the 6 month option is selected, data from 6 months ago to date will be sent to this organization's domain. After the selections are made, the query corresponding to these selections is created in the backend and the data is extracted from the PostgreSQL database in csv format. Then this csv is added to the private IPFS network via the API port of the IPFS nodes. The hash obtained from IPFS is passed to react in order to be used in the IPFS interface to let users download data [32]. Data sharing stages are given in **Figure 4**.

When a third party wants data from our platform, this is the first screen they will see (**Figure 5(a)**). The screen shows information on how we get data. Then they specify the organization, domain, subdomain and time range of data (**Figure 5(b)**). After selections are made by clicking on the IPFS icon, the user can request the data (**Figure 5(c)**). In the last screen (**Figure 5(d)**), the hash in the red block is passed to the React front-end and the user can download the data in CSV format from the IPFS interface. **Figure 5(d)** presents the IPFS Interface. Exposing the API port is not recommended as it provides admin-level access to the node, so the API port is kept local and only accessible through our Node.js backend. A gateway is exposed to the public internet to let users download content from our private network via an IPFS interface designed for browsers.

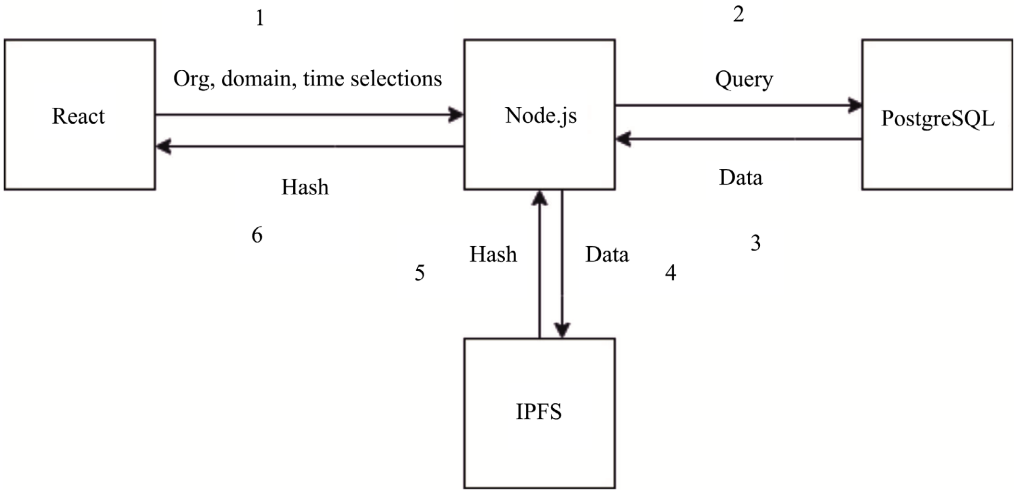

Figure 4. Data sharing stages.

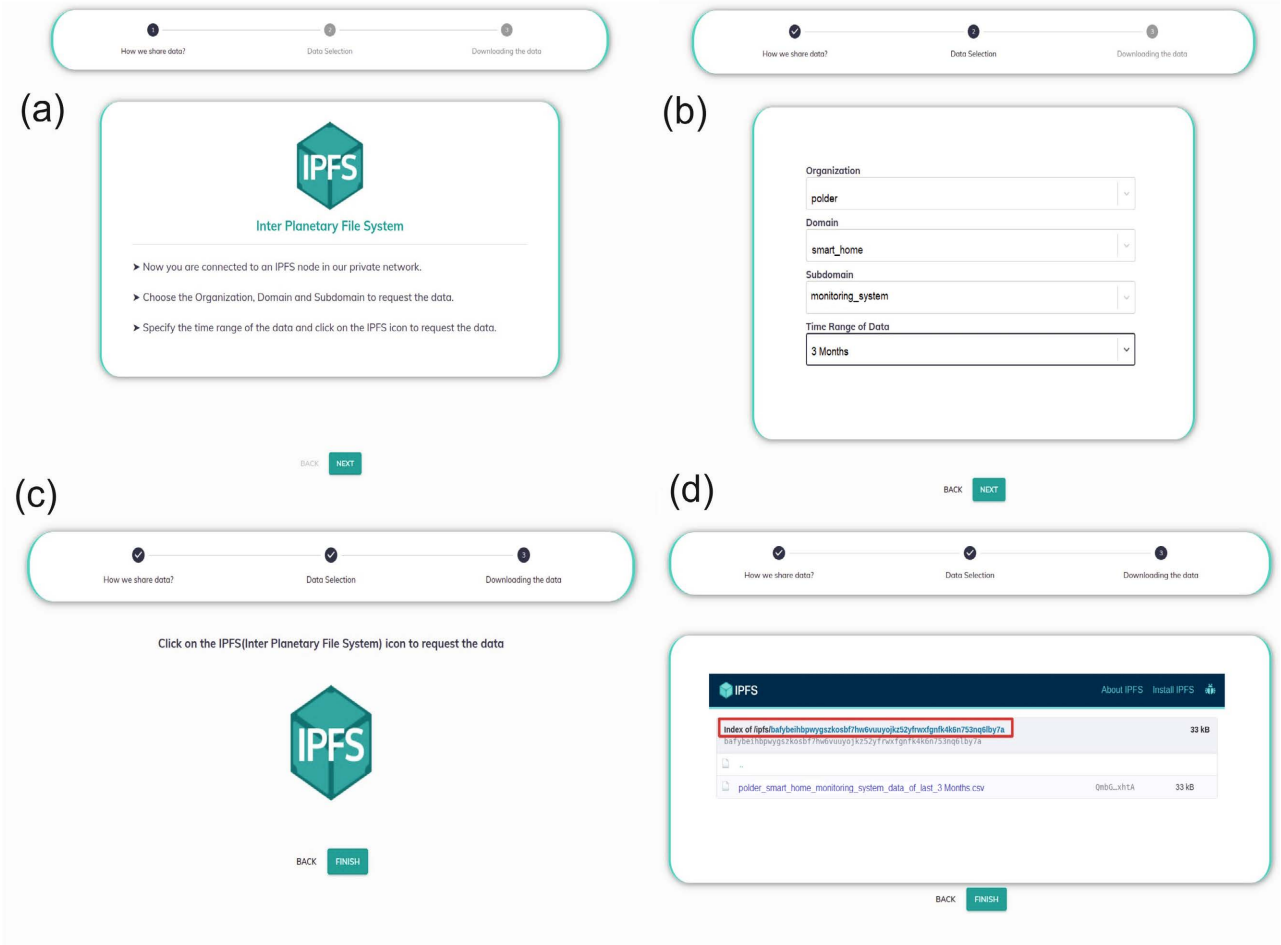

Figure 5. (a) Information on how we get data (b) selecting the data (c) confirming the selection (d) downloading data from IPFS Interface.

On the public network, everyone can access the content added by at least one of the nodes in the global network. Nowadays, approximately 800 nodes are online in public IPFS. In a private network, nodes that share the same swarm key are connected to each other. The content added to the private IPFS network is only reachable by the nodes inside the private IPFS network. If a node in the network knows the hash of a specific file, it can request the file from the network. We currently have 2 nodes. The count of nodes can be scaled according to the availability needs of the information.

IPFS uses Content Addressing to access data. Also, IPFS takes any size or type of data and gives a fixed-length, 46-byte hash. IPFS hashes have some important characteristics. These are given in Table 1.

3.1.3. Security

In the development environment, the MQTT protocol is used for getting sensor measurement data and HTTP is used for communication between micro services. But in a production environment, HTTP and MQTT protocols are not secure as their traffic is not encrypted and can be intercepted by third parties. As a

Table 1. Some important characteristics for IPFS hashes.

	Explanations
Deterministic even in different private networks	IPFS gives the same hash if the the same file is added to IPFS
Uncorrelated	Even a small change in a file means a completely different hash
Unique	It's impossible to obtain the same hash from two different files
One-way	It's impossible to guess or calculate the input message from its hash

result, running Keyrock Idm, Wilma Pep Proxy, and Orion Context Broker in HTTPS mode is preferred, and MQTTS is used to retrieve sensor data. All data traffic from sensors to the Orion Context Broker will be encrypted. User and password authentication is also used in MQTT. Another problem to take into consideration is that users can publish and subscribe to all topics in MQTT if they have a valid username and password and the client certificate created by us. In order to prevent a user from listening to another user's topics, we create Access Control Lists in MQTT. This way, users can only listen to their own topics and can't publish or subscribe to other users' topics. After adding a user and password to MQTT and adding new things to the Access Control List, we need to restart MQTT for these updates to happen. Doing these updates in real time via a dynamic plugin is still under development.

3.2. Sensor and Device Registrations

A device is added to the system from the API testing tool Postman. For example, a device named "PositionMeasuringDevice" is added in smart_home and monitoring_system named fiware-service and fiware-servicepaths. Then, "mykey" is given as a security key to this device and a sensor named "dx_position" is added to PositionMeasuringDevice. This step is adding a sensor to previously registered devices. A trust token is taken from Keyrock Idm by supplying the user's login credentials, which is getting a trust token from keyrock added to the device, and the address of PEP Proxy is defined. Now with each sensor data coming to Orion Context Broker, the device will be authenticated at Pep Proxy using this trust token. Giving devices a specific time limit that they can send data and denying the access afterwards is still under development. From now on, the data coming from this sensor will be published to "/*mykey*/*dx_position*/*attrs*" topic by the user. In Figure 6, value 250 is published to this topic using MQTT.

It is seen that the IoT Agent carried this measurement to a related part of the sensor in the MongoDB database by subscribing the topics according to the security keys of the devices, and the latest sensor measurement is obtained. Python 3.6, JavaScript, and SQL are among the project programming languages.

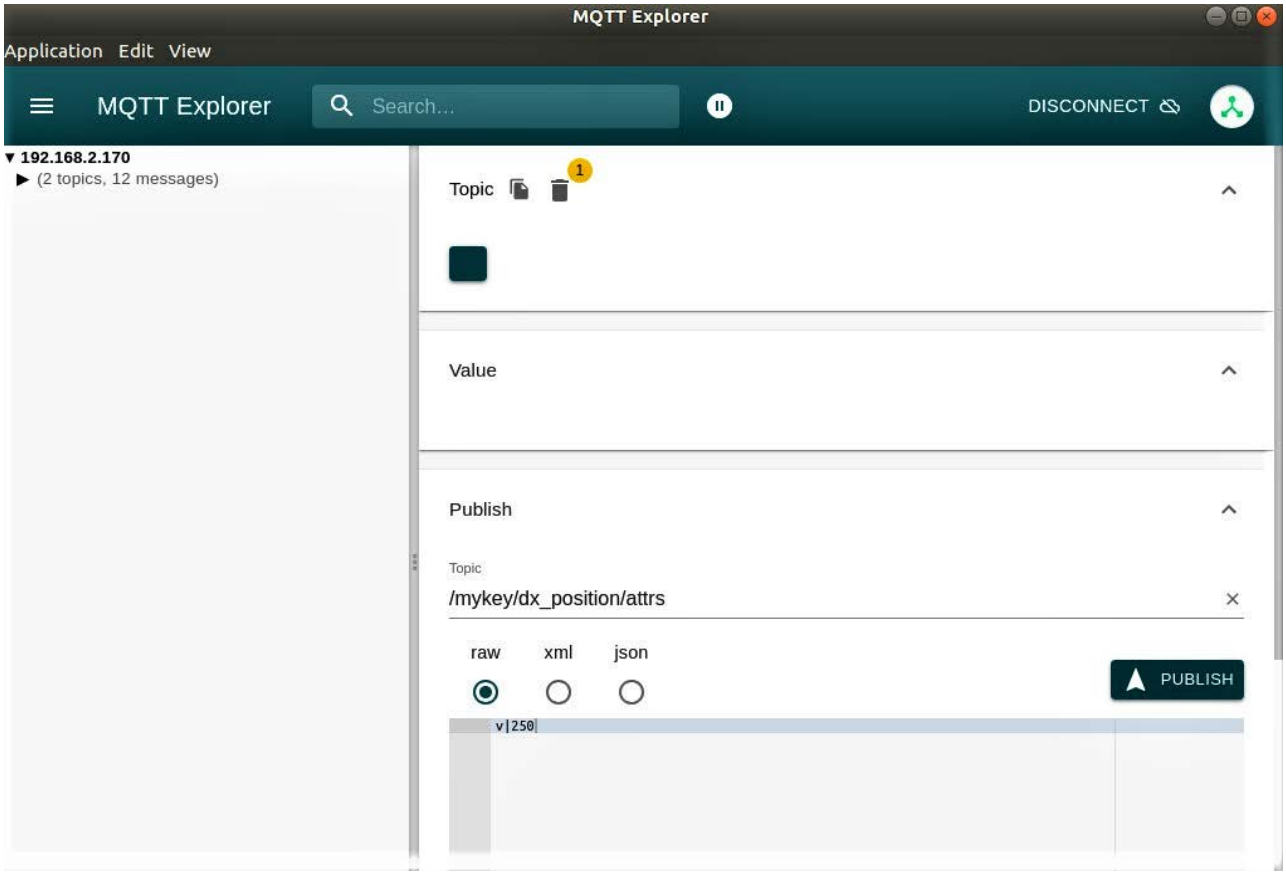

Figure 6. Publishing a value in /securitykey/sensorid/attrs format.

3.3. Identity Management

In our monitoring system, Keyrock Generic Enabler is responsible for identity management. Keyrock provides OAuth2-based authentication and authorization security to the system. Keyrock uses a MySQL database to store user identities, roles, and permissions. To ensure that only registered users are served, we deny access to data sent from unregistered sources. Users have no direct communication with the context broker due to the Wilma Pep proxy being in the middle of the user and the context broker. Therefore, security is reinforced by adding one more layer of authentication and authorization. Wilma Pep proxy uses the standard OAuth2 protocol. When a request is sent, it is checked whether the access token in the request header is valid. According to the validation result, the request is either allowed or denied.

3.4. Back End

3.4.1. NodeJS

NodeJS is used to communicate with React and Python easily [33]. With React UI, the information received from the user is transferred to the route specified on the Express service on NodeJS. Then, this information is converted into json data and sent as an http request by adding the dynamic fiware-service name to the end of the specific route on the flask using the axios module. This informa-

tion includes: sensor name, fiware-service and fiware-service-path, the time interval, and duration (h).

3.4.2. Flask

In the system, the purpose of configuring flask [34] is to communicate with the routes configured on flask and express routes on NodeJS, and also to gather data from the PostgreSQL database where Fiware collects its data. A dynamic URL structure works in both Node.js and Flask. The fiware-service selected by the user is added to the end of the URL. In this way, if two processes are triggered at the same time in the python ecosystem, the flask creates parallel routes with the multiprocessing methodology.

3.5. Energy Consumption Calculation

The electrical energy used by the electrical device is the product of the electrical power (P) and the time elapsed (t) (3).

$$E = P \times t \tag{1}$$

Kilowatts (kW) and kilowatt-hours (kWh) are calculated according to the formula below (4). This product is divided by 1000 to get the result in kilowatt-hours.

$$\text{kWh} = \text{Watt} \times \text{time}\left(\text{hours}\right)/1000 \tag{2}$$

For example, if we use a washing machine that requires power consumption of 303 watts (0.303 kW) and the washing machine runs for 1 hour, it will consume 0.30 kWh of electrical energy. If we use it for 2 hours per day, then this washing machine will consume electricity per day at 0.60 kWh, or 18 kWh per month. The energy consumption is automatically stored and calculated in a database on the server performing the calculation in Equations (1) and (2). To calculate how many Euros the electricity consumption will be equal to, the total electricity consumption of a device (kWh) is multiplied by the specific unit price of electrical energy (€). The user can share the electricity consumption and related cost data with a third party via IPFS. This data is collected from the household devices registered to the monitoring system for 1 month, 3 months, 6 months, and 1 year and stored in the PostgreSQL database in .csv format. Energy-related data can be used by an institution in AI research and development studies, such as a smart home energy management system.

3.6. Sensors

The ACS712 wireless sensor, which operates on the Hall Effect principle, is used in the proposed monitoring system to measure the flow of electricity (current and voltage) to a smart home household device. They are highly effective sensors for metering and measuring the overall power consumption of systems [35]. The sensor can read and control a maximum current of up to 5 amps. All the data that is being measured can be transmitted to a remote monitoring system. The sensors were bought commercially. With the wireless sensor, we measure and

record the energy consumption of a household device and determine the total energy consumption. The sensor can communicate with the IoT platform during the sending of energy consumption data and provide a low-cost solution. The main disadvantage of the sensor is that calibration is required for more accuracy because of its low cost. The most important feature is that it can function between −40˚C and 85˚C and give accurate current measurement for both AC and DC signals. The data coming from the sensor is published to the MQTT Broker (ACD Server). After that, the broker publishes a second time for subscribers to the project. The client is an ACD database, receiving data on a specific topic. Home appliances all operate in distinct ways. Dishwashers, vacuum cleaners, televisions, washing machines, hair dryers, cookers, irons, etc. have two statuses (open and closed). For appliances with two statuses, the running time can be changed (time-shift load).

3.7. Load Tests

In the load tests, a total of 1000 http requests were published on Orion CB over 2 fiware-services and 17 fiware-service paths. These requests are in the NGSI-v2 format accepted by Orion, and the Batch update or Create entity method was used while publishing the request. The published sensor values were set to be random. Python (a programming language) was used to generate synthetic data, and Newman CLI, supported by Postman, was used for load testing. Figure 7 shows the results of the load tests performed.

4. Experimental Results and Discussion

The user interface facilitates the usage of the IoT platform without requiring software knowledge. In this study, ReactJS was used to design such an interface that can be divided into two tasks. The former is user-prompted specifications such as electricity costs per hour for a specific country, a usage time of the device, sensor data, fiware-service and fiware-service path information. The latter is a dashboard for providing information to the user after processes are done. In Figure 8, at the timestamps between 00:00 and 08:00, the cost is 0.139 € with a unit of kW/h.

	executed	failed
iterations	1000	0
requests	17000	0
test-scripts	0	0
prerequest-scripts	0	0
assertions	0	0
total run duration: 5m 30.2s		
total data received: 125kB (approx)		
average response time: 7ms [min: 1ms, max: 91ms, s.d.: 3ms]		

Figure 7. Load test results.

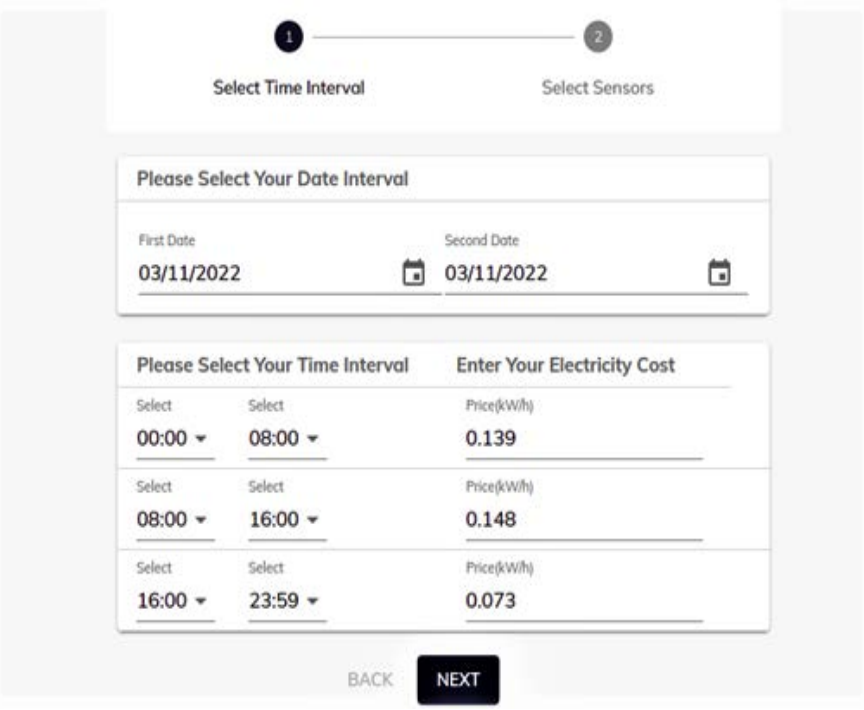

Figure 8. Electricity costs per hour for a specific country.

The user can add any number of devices to the home energy monitoring system. The system shows the energy consumption of each sensor (kWh), total energy consumption of all sensors (kWh), unit electricity cost per hour (€), the electricity cost of each sensor (€), and the total electricity cost of all sensors (€). The energy sensor node takes the energy consumption and sends it to the server. Sensors reflect their own ID and send data every hour (frequency) throughout the day.

4.1. Energy and Cost Monitoring

In the energy monitoring system, the data on energy consumption collected from the sensors (per 5 min) is displayed on the dashboard to the user. The system shows the energy consumption (kWh) for household devices over a given time period. In the dataset, the electricity cost is determined as three tariffs in a day, and the unit is Euro (€). The organization name (energy_monitoring), domain name (smart_home), and sub-domain name (energy) can be selected from the general menu to monitor energy and cost data. The general menu also has "Graphs Type Selection", including area, table (raw data), and chart graphs. The data collected from the sensors is directly reflected on the screen.

Figure 9 shows the cost change (€) on users' electricity consumption in a day. The user selects "chart" and "cost values" obtained "cost change" from the general menu. The user also can get the minimum (min), maximum (max) and average (avg) values of energy consumption in a day by choosing "area graph". Figure 10 also shows the total energy (kWh = kW * hour) consumed by each device and all devices.

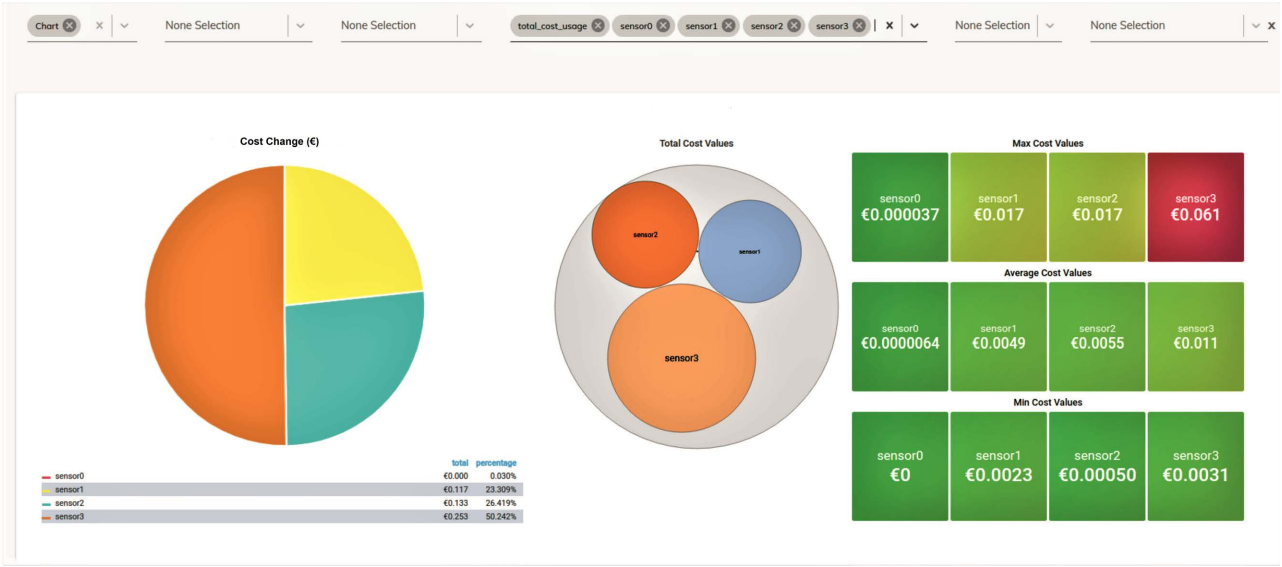

Figure 9. The cost change (€) on users' electricity consumption for each sensor.

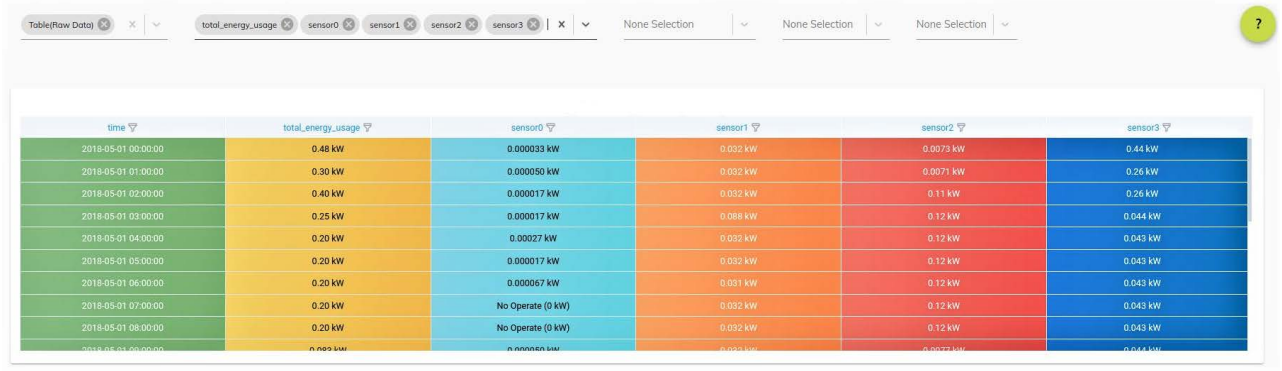

Figure 10. The total energy consumed by each sensor and all sensors.

4.2. Discussion

A private IPFS network is used in the energy monitoring platform to increase system security. The energy consumption and total cost data for household devices are measured and recorded using wireless sensors. A third party requesting data is able to download from the organizations that permit data sharing. This data leads to greater quality and R & D in energy efficiency technologies for home devices.

Comparison with the Existing Works

In this section, we compare the proposed work with the existing works. Gunturi and Reddy [9] developed a device to measure the kWh readings of specific appliance and integrated this energy metering into IoT. The user can view his or her app's energy consumption readings. The app downloads data from the cloud and shows it in the desired format—lists or graphs. In our study, data sharing on our platform takes place over a secure IPFS network. A third party can download all the energy consumption and cost data for each device from the platform

for 1 day, 3 months, 6 months, and 1 year. Kumaresan *et al.* [35] used a smart energy meter to measure and analyze the power consumption for smart home appliances. The hardware components used to measure the electrical current include the Arduino UNO, ESP8266, 4 relay modules, and an ACS712 hall sensor. Similarly, the ACS712 hall sensor was used to measure the flow of electricity to a device (see Section 3.7) in our monitoring system. Compared to their work, our system based on IoT can be expanded to larger homes by expanding the number of sensors and measuring parameters.

5. Conclusions and Future Work

The research project presents a home energy monitoring system that shows the energy consumption and excess consumption of household devices. The analysis quickly identifies the home appliance usage pattern. This pattern discovery can control the unnecessary operation of the device, and then the user can take appropriate action on it. The resident can conveniently monitor kWh measurements. Every home may save energy to limit the amount of electricity that is wasted. Any number of devices can be added to the monitoring platform. In this study, data sharing on the platform flows over a secure IPFS network. Over this secure network, a third party can download all energy consumption and cost data from the platform for each device. It also helps manufacturers make their products more efficient.

There has been a significant increase in the amount of electricity used by household equipment. To meet demand, the monitoring system will be changed into a home energy management system in the future, taking into account what users want. We also plan to extend this platform by putting our solution on an open-source blockchain platform like Ethereum or the Hyperledger Blockchain.

Acknowledgements

The authors would like to thank ACD Bilgi İşlem Bilgisayar Yazılım Hizmetleri San ve Tic. Ltd. Sti. for its contribution to the study.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Compliance with Ethical Statement

This submitted work is original and has not been published elsewhere.

References

[1] Suciu, G., Necula, L., Iosu, R., Usurelu, T. and Ceaparu, M. (2019) IoT and Cloud-Based Energy Monitoring and Simulation Platform. 2019 11*th International Symposium on Advanced Topics in Electrical Engineering*, Bucharest, 28-30 March 2019, 1-4. https://doi.org/10.1109/ATEE.2019.8724961

[2] Karthick, T., Charles Raja, S., Jeslin Drusila Nesamalar, J. and Chandrasekaran, K. (2021) Design of IoT Based Smart Compact Energy Meter for Monitoring and Controlling the Usage of Energy and Power Quality Issues with Demand Side Management for a Commercial Building. *Sustainable Energy, Grids and Networks*, **26**, Article ID: 100454. https://doi.org/10.1016/j.segan.2021.100454

[3] Velasquez, W., Tobar-Andrade, L. and Cedeno-Campoverde, I. (2021) Monitoring and Data Processing Architecture Using the FIWARE Platform for a Renewable Energy Systems. 2021 *IEEE* 11*th Annual Computing and Communication Workshop and Conference*, Nevada, 27-30 January 2021, 1383-1387. https://doi.org/10.1109/CCWC51732.2021.9376026

[4] Santos, D. and Ferreira, J.C. (2019) IoT Power Monitoring System for Smart Environments. *Sustainability*, **11**, Article 5355. https://doi.org/10.3390/su11195355

[5] Sanz, R., Corredera, Á., Hernández, J.L., Samiengo, J., Vicente, J.M. and Bujedo, L.A. (2015) Towards the Integration of Monitoring Systems to Support the Evaluation of Nearly Zero Energy Buildings through Key Performance Indicators. 2015 *IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems*, Trento, 9-10 July 2015, 90-95. https://doi.org/10.1109/EESMS.2015.7175858

[6] Pellegrino, A., Lo Verso, V.R.M., Blaso, L., Acquaviva, A., Patti, E. and Osello, A. (2016) Lighting Control and Monitoring for Energy Efficiency: A Case Study Focused on the Interoperability of Building Management Systems. *IEEE Transactions on Industry Applications*, **52**, 2627-2637. https://doi.org/10.1109/TIA.2016.2526969

[7] Ruiz, L.G.B., Rueda, R., Cuéllar, M.P. and Pegalajar, M.C. (2018) Energy Consumption Forecasting Based on Elman Neural Networks with Evolutive Optimization. *Expert Systems with Applications*, **92**, 380-389. https://doi.org/10.1016/j.eswa.2017.09.059

[8] Tanasiev, V., Pătru, G.C., Rosner, D., Sava, G., Necula, H. and Badea, A. (2021) Enhancing Environmental and Energy Monitoring of Residential Buildings through IoT. *Automation in Construction*, **126**, Article ID: 103662. https://doi.org/10.1016/j.autcon.2021.103662

[9] Gunturi, S.K.S. and Reddy, M.S.K. (2018) IoT Based Domestic Energy Monitoring Device. 3*rd International Conference for Convergence in Technology*, Pune, 6-8 April 2018, 1-4. https://doi.org/10.1109/I2CT.2018.8529756

[10] Gan, S., Li, K., Wang, Y. and Cameron, C. (2018) IoT Based Energy Consumption Monitoring Platform for Industrial Processes. 2018 *UKACC* 12*th International Conference on Control*, Sheffield, 5-7 September 2018, 236-240. https://doi.org/10.1109/CONTROL.2018.8516828

[11] Nord, J.H., Koohang, A. and Paliszkiewicz, J. (2019) The Internet of Things: Review and Theoretical Framework. *Expert Systems with Applications*, **133**, 97-108. https://doi.org/10.1016/j.eswa.2019.05.014

[12] Onay, A., Akın, Y., Kafalı, A. and Çıracı, E. (2021) Real Time Air and Water Quality Monitoring Based on Distributed Sensor Network. *2021 6th International Conference on Computer Science and Engineering* (*UBMK*), Ankara, 15-17 September 2021, 118-123. https://doi.org/10.1109/UBMK52708.2021.9558881

[13] Chooruang, K. and Meekul, K. (2019) Design of an IoT Energy Monitoring System. *International Conference on ICT and Knowledge Engineering*, Bangkok, 21-23 November 2018, 1-4. https://doi.org/10.1109/ICTKE.2018.8612412

[14] Luan, H. and Leng, J. (2016) Design of Energy Monitoring System Based on IOT. *2016 Chinese Control and Decision Conference* (*CCDC*), Yinchuan, 28-30 May 2016, 6785-6788. https://doi.org/10.1109/CCDC.2016.7532219

[15] Benet, J. (2014) IPFS—Content Addressed, Versioned, P2P File System. http://arxiv.org/abs/1407.3561

[16] Ali, M.S., Dolui, K. and Antonelli, F. (2017) IoT Data Privacy via Blockchains and IPFS. *Proceedings of the Seventh International Conference on the Internet of Things*, Linz Austria, 22-25 October 2017, 1-7. https://doi.org/10.1145/3131542.3131563

[17] Krco, S., Pokric, B. and Carrez, F. (2014) Designing IoT Architecture(s): A European Perspective. 2014 *IEEE World Forum on Internet of Things*, Seoul, 6-8 March 2014, 79-84. https://doi.org/10.1109/WF-IoT.2014.6803124

[18] Samuel, S.S.I. (2016) A Review of Connectivity Challenges in IoT-Smart Home. 2016 3*rd MEC International Conference on Big Data and Smart City*, Muscat, 15-16 March 2016, 1-4. https://doi.org/10.1109/ICBDSC.2016.7460395

[19] Lloret, J., Tomas, J., Canovas, A. and Parra, L. (2016) An Integrated IoT Architecture for Smart Metering. *IEEE Communications Magazine*, **54**, 50-57. https://doi.org/10.1109/MCOM.2016.1600647CM

[20] Alaa, M., Zaidan, A.A., Zaidan, B.B., Talal, M. and Kiah, M.L.M. (2017) A Review of Smart Home Applications Based on Internet of Things. *Journal of Network and Computer Applications*, **97**, 48-65. https://doi.org/10.1016/j.jnca.2017.08.017

[21] Fletcher, J. and Malalasekera, W. (2016) Development of a User-Friendly, Low-Cost Home Energy Monitoring and Recording System. *Energy*, **111**, 32-46. https://doi.org/10.1016/j.energy.2016.05.027

[22] Tran, L.N., Gao, W. and Ge, J. (2021) Sensitivity Analysis of Household Factors and Energy Consumption in Residential Houses: A Multi-Dimensional Hybrid Approach Using Energy Monitoring and Modeling. *Energy and Buildings*, **239**, Article ID: 110864. https://doi.org/10.1016/j.enbuild.2021.110864

[23] Al-Kuwari, M., Ramadan, A., Ismael, Y., Al-Sughair, L., Gastli, A. and Benammar, M. (2018) Smart-Home Automation Using IoT-Based Sensing and Monitoring Platform. 2018 *IEEE* 12*th International Conference on Compatibility, Power Electronics and Power Engineering* (*CPE-POWERENG* 2018), Doha, 10-12 April 2018, 1-6. https://doi.org/10.1109/CPE.2018.8372548

[24] Srinivasan, A., Baskaran, K. and Yann, G. (2019) IoT Based Smart Plug-Load Energy Conservation and Management System. 2019 *IEEE* 2*nd International Conference on Power and Energy Applications*, Singapore, 27-30 April 2019, 155-158. https://doi.org/10.1109/ICPEA.2019.8818534

[25] Ma, X. and Luo, W. (2008) The Analysis of 6LowPAN Technology. 2008 *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, 19-20 December 2008, 963–966. https://doi.org/10.1109/PACIIA.2008.72

[26] Wang, X. (2015) Multicast for 6LoWPAN Wireless Sensor Networks. *IEEE Sensors Journal*, **15**, 3076-3083. https://doi.org/10.1109/JSEN.2014.2387837

[27] Conde, J., Munoz-Arcentales, A., Alonso, A., Lopez-Pernas, S. and Salvachua, J. (2021) Modeling Digital Twin Data and Architecture: A Building Guide with FIWARE as Enabling Technology. *IEEE Internet Computing*, **26**, 7-14.

[28] Fiware-Orion. (2022) Welcome to Orion Context Broker.

[29] (2022) MongoDB. https://www.mongodb.com

[30] (2022) Fiware Draco. https://fiware-draco.readthedocs.io/en/latest/

[31] PostgreSQL (2022) The World's Most Advanced Open Source Database.

[32] (2022) React. https://reactjs.org/

[33] Huang, X. (2020) Research and Application of Node.js Core Technology. 2020 *In-*

ternational Conference on Intelligent Computing and Human-Computer Interaction (*ICHCI*), Sanya, 4-6 December 2020, 1-4.
https://doi.org/10.1109/ICHCI51889.2020.00008

[34] Yaganteeswarudu, A. (2020) Multi Disease Prediction Model by Using Machine Learning and Flask API. 2020 5*th International Conference on Communication and Electronics Systems* (*ICCES*), Coimbatore, 10-12 June 2020, 1242–1246.
https://doi.org/10.1109/ICCES48766.2020.9137896

[35] Kumaresan, P., Prabukumar, M. and Barathkumar, E. (2020) SMART HOME: Energy Measurement and Analysis. 2020 *International Conference on Emerging Trends in Information Technology and Engineering* (*ic-ETITE*), Vellore, 24-25 February 2020, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.ICETITE318