Home Automation Architecture based on IoT Technologies

Judson Costa, Daniel Araújo, Diego R. C. Silva, Marcelo B. Nogueira and Marconi C. Rodrigues
Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte 59078-275
Emails: judson.scosta@bct.ect.ufrn.br, danielaraujo@bct.ect.ufrn.br,
diego@ect.ufrn.br, marcelonogueira@ect.ufrn.br, marconicamara@ect.ufrn.br

***Abstract*—This article proposes a Home Automation Architecture based on IoT Technologies. The proposed architecture allows the storage of domiciliary data and the achievement of strategies to better manage the resources, in order to seek greater efficiency in energy consumption and allow a sustainable model. The architecture is built from open platforms and results in an effective solution that allows bi-directional data flow with low latency, fully compatible with the Wi-Fi network, it works with the local network and, consequently, there is the possibility of use in environments that do not have an internet connection.**

I. Introduction

Home automation uses a variety of technologies to provide a different lifestyle than it is currently seen. The progressive development of the computational electronics infrastructure has led to the development of servers with competence for intensive data accumulation and management, known as big data [1], opening doors to the concept of IoT, the internet of things [2], that transforms simple objects into intelligent devices [3]. In the scope of wireless communication, several specifications were developed for sending information between electronic devices with the objective to increase safety, signal range and many other aspects. Some of the existing protocols are Z-Wave; Bluetooth; Zigbee; Wi-Fi. Communication protocols have also evolved to meet, for example, the need for low response time; new security standards in data transfer and; real-time communication. Hypertext Transfer Protocol Version 2 (HTTP / 2), for example, was designed to provide more efficient use of network, data compression, and other enhancements [4], and with the addition of the SSL / TLS security layer results in the HTTPS [5] protocol, which encrypts the client-server connection, making it difficult to view the data transferred by third parties.

This article will present an architecture for residential automation that facilitates the installation of intelligent devices without complications and technical knowledge from a web application and a central server as Web Service connected to all the devices of the residence. This architecture adopts: Wi-Fi protocol - because it is a wireless communication network that has several advantages such as mobility, compatibility, sharing, accessibility, low maintenance cost, security, and popularity; the WebSocket and MQTT protocol - for real-time communication and bi-directional connections to send and receive messages with low latency; and the HTTPS protocol - for data transfer with an encrypted connection.

With the proposed architecture, end users connected to the system will enjoy several benefits, such as: mobility, the user can control their devices regardless of distance; security, the user can create rules in the system that is able to inform about events happened in his residence and already take providences triggering devices; maintenance, easy monitoring of their devices, contributing to corrective maintenance; control, the user can change the status of the devices in real time.

II. Architecture

The architecture uses a set of sensors and actuators and can be installed in a residence in such way that they can communicate with each other, being possible through a web application, available in a central unit and accessible by computer or smartphone connected to the same network, It is possible to manage any smart device that is registered, and customize the access permission in a simplified way, like customize the creation of levels that can represent, for example, the sectors of a company or the rooms of a residence, such as a tree hierarchy. In addition, a user can actuate in devices manually or automatically by configuring a set of rules from sensor data, as well as view the data from sensors and create groups of devices that can be triggered together.

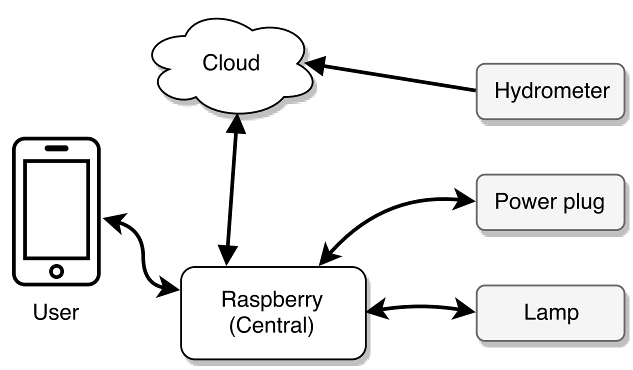

Figure 1. Architecture

As shown in Figure 1, the solution is composed of a central unit (middleware) that through the Wi-Fi network of a residence connects to intelligent devices, sensors, and actuators, for monitoring a place, such as energy and water consumption measurement, and activation of power plugs, lighting, and air-conditioning systems.

A. *Web development*

Web development is a widespread technology, easy to access and easier to implement compared to native desktop or mobile programming, such as C/C++ or Java, as well as being cross-platform, allowing the web application to run on any operating system that supports a browser, regardless of whether it is a desktop or smartphone. This results in a simpler maintenance than a dedicated application for each of the systems and devices on the market since web application support will depend only on the browser version available.

For the development of web interfaces, the React.js library was chosen because it is efficient and based on components. For the creation of the application executed in the central unit, it was used the Node.js, which is a framework for the construction for high performance and asynchronous flow applications obligatorily that allows to receive multiple connections and to process them simultaneously [6]. The two implementations are written using the JavaScript language.

In addition to the software, it was necessary to select components for hardware that offer a complete experience. Among them, the Raspberry PI 3 is used as a central unit and local server because it offers a satisfactory relation of processing and cost, wireless internet connection, and a reduced size; and the ESP8266 enables the use of a microcontroller with low dimensions, high processing capacity, and Wi-Fi access intrinsic to the chip, in addition to low consumption. All of these devices communicate on the web with Wi-Fi and the application protocols HTTP(S), WebSocket and MQTT.

B. *Central unit*

The main component of this architecture is the central unit, the middleware, that mediates between the web interface accessed by the users and all the intelligent devices connected to a residential network. In addition to allowing the control without the user being obligated to be connected to the internet, except to access externally. The central is responsible for: managing all user interface requests; processing the rule block; storage of all data generated by the sensors and; actuation of the actuators by users and/or rule block. The central unit is a Web Service that receives data in a predefined JSON format and returns the requested information. By standardizing this data input and output format it is possible to communicate with any compatible smart devices and interfaces.

The middleware consists of several tools and libraries that work together and are separated into containers, as shown in Figure 2. The application follows the principles of the Representational State Transfer (RESTful) standard that uses HTTP [7] methods to facilitate the external access of devices to the central unit, which receives the generated data, for example, water consumption from water meters; energy consumption from power plugs; recording the activation of lighting, power plugs and/or air-conditioning system.

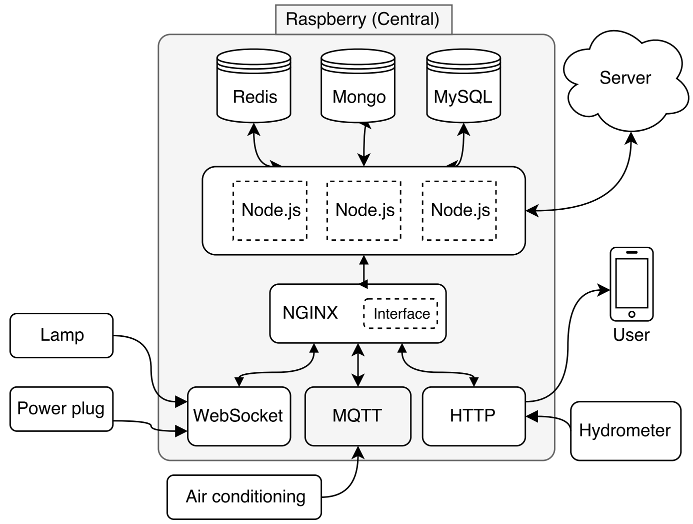

Figure 2. Tools used in the Central Unit

The user, via the web interface, and the various intelligent devices, connect to the central unit through HTTP(S), WebSocket or MQTT protocols. The choice between the supported protocols depends on the volume of data and the need for real-time monitoring or not. The requests are processed by the application running on Node.js. All the logic of reading and writing data, executing the rule block, connecting to the cloud to, if necessary, external access and communication with the three types of databases is done by Node.js.

It is known that the storage of the data is of paramount importance, but to file them safely it is necessary to know the format and priority of processing. In this architecture, three different databases are used for different purposes of processing and each one has its advantages and disadvantages when comparing their respective performances [8].

1) Redis: It is a non-relational database that stores data in memory and has a high read-write performance compared to the Database Management System (DBMS). It is used to store data that need quick access, such as the current situation of the sensors and actuators present in a residence, in addition to enabling the clusters of the Node.js application and the WebSocket connections.

2) MongoDB: It is also a non-relational database, however, it uses disk storage. Even so, it is excellent for storing a large amount of data, such as records of water metering; consumption, among others.

3) MySQL: With disk storage, MySQL is a relational DBMS that performs well in data indexing, however, it does not allow horizontal scaling [8]. MySQL is used to store the data of users, devices, groups and their relationships since it guarantees that a group can only be registered if it is created by an active user and has some linked device.

C. Devices

The devices used are all developed based on the ESP8266 microcontroller (MCU) due to the chip supported Wi-Fi connectivity and security standards such as WEP, WPA-PSK, WPA2-PSK and WPS [9] that allow direct integration between sensors and actuators with a local server. These properties of this electronic component satisfy the needs of this architecture due to the environment in which it is inserted to be based on web communication protocols.

The use of an MCU in devices makes the range of sensors and actuators that can be installed for automation of general tasks such as: crop control in agriculture [10]; lighting systems and energy efficiency control [11], among others. Because of this, the possibility of different ways of using the same microcontroller in function of the set of rules created by the user can be exemplified the automation of simple tasks such as monitoring and rapid activation. One of the situations can be when adding a motion sensor and a group of intelligent lights, a business rule that with the detection of some movement in the local by the sensor turns on / off that same group of lamps, besides allowing the configuration of the time that the lights should remain on.

The devices are a set of sensors and actuators where the actuators receive data from the central unit for its operation and also report their status to the central unit if it has some external interference. The sensors send data to the central unit to be stored in the history for later viewing in the online interface. The device can also generate error logs if there is any difficulty in its full functioning that can also be viewed online.

D. Integration

The devices connect to the central unit through the home Wi-Fi network and use HTTP(S), MQTT and/or WebSocket protocols. The choice of which protocol to use depends mainly on the response time required in data transfers, for example, the action of activating a certain lamp and the sending of a measurement performed by a water meter have different priorities, while the first one requires instant action, the second one doesn't need to have an immediate response. All devices, in addition to reporting the data obtained in the monitoring, also report to the central unit when it identifies some abnormality in its own functioning, such as problems sending data, connecting to the internet or unexpected restart. Devices and users are authenticated following the OAuth 2.0 [12] specification that standardizes authentication through temporary security keys.

The Figure 3 assembles a scenario to show how the devices, the user, the central unit, and the cloud communicate. Intelligent devices (power socket, water meter, and lamp) communicate with the central unit or the cloud, receiving actions to be performed and sending monitored data. Due to the low level of urgency of response, the water meter is the only one that connects directly to the cloud and uses only the HTTP protocol, represented by "POST", which is one of the methods of this protocol, to monitor water consumption and send the data to the server. Unlike the water meter, the user interface, the power plug, and the lamp have a higher level of response time, requiring a return immediately after the request. To meet this need, both devices connect directly to the central unit via the WebSocket or MQTT protocol for real-time communication. In addition to managing all these connections with different devices and protocols, the middleware also connects to the cloud in specific cases where the request asks some information that is not found locally or to allow external access to the user's home, which is optional.

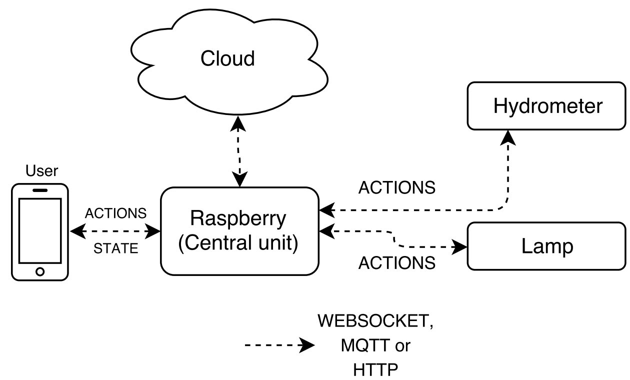

Figure 3. Communication between devices

III. RESULTS AND DISCUSSIONS

Initially, this architecture only allowed the registration of devices that were compatible with the supported types, this restricted the creation of new devices that had a small variation of functionality. For example, you can compare a power meter and a water meter, they are devices with different names and applications but have the same functionality of measuring something. With this in mind, this architecture allows to dynamically describe an intelligent device from the functionalities offered by it, that are compatible with the functions supported by this architecture. The platform creates one type per device, allowing multiple sensors and actuators using one accepted protocol, the groups of devices use common aspects between the devices to aggregate functionalities. Each sensor or actuator have their one id it uses to communicate to the middleware, with this functionality, the device don't need to post its complete value, just need to send the value of the sensor that changed.

A. Comparison

There are architectures that share proposed architecture aspects in this article.

Improving Heterogeneous SOA-Based IoT Message Stability by Shortest Processing Time Scheduling [13] proposes a three-layer architecture implementing the broker (analogous to the central unit) using the HTTP protocol through the polling process to schedule the process time to optimize the operation of your server, but there are other communication protocols that could solve the exchange of information problem in real time.

The Social internet of things [14]: uses only the HTTP and applies the concept of a social network between the devices mimicking the relationship between humans, but in practice, this relationship is done by the end user creating groups and relate them to rules that allow one device communicates directly with another device without interference from the server or central unit. This functionality does not work in case of multiple sensors generating a single value for a device so the architecture described in this paper uses the central unit to process the sensors input data and triggers the rules made by the users.

Most of the commercial applications surveyed like Control4® show closed protocols with no possibility of communication between architectures, some use third party API like Philips Hue Lights® or Xiaomi Yeelight® to expand the number of accepted devices but do not open their own API for external control.

B. Interface

The application can simulate a native application from the default mobile operating system browser. Network traffic has been reduced to the maximum because the interface saves data that has already been requested from the server.

The interface uses a user interface framework called material-ui as a set of components using the material design, that is a google standard design.

The figure 4 shows the web application interface that provides the history of the water consumption of the device connected to the intelligent water meter. The interface allows you to view the history of smart devices registered and the sampling period with daily, monthly and annual resolution. The visualization of figure 5 shows the error logs recorded by the device when there is any interference with the normal operation of the device.

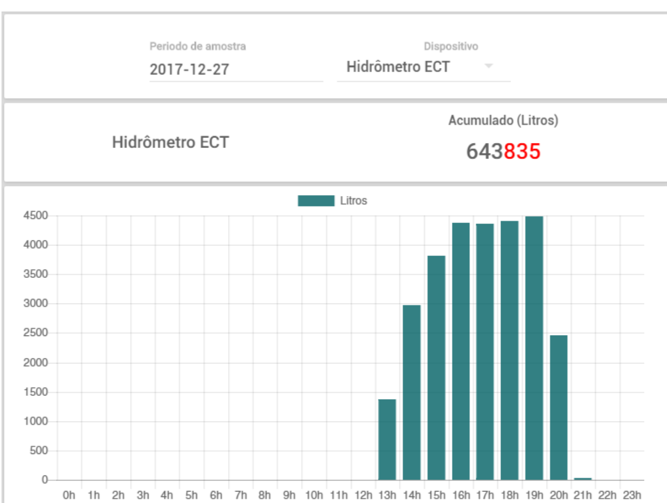

Figure 4. Interface screen showing graph of water consumption of a smart water meter

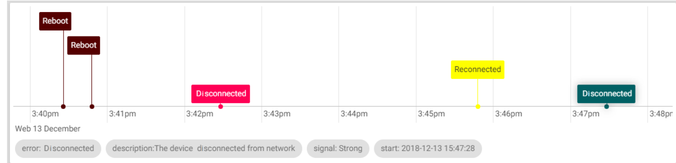

Figure 5. Interface screen showing the error logs of a smart water meter

C. Architecture validation

To validate the proposed architecture, some prototype devices were assembled to communicate with the central unit to check the response time of the web interface, the central unit, and the connected device itself. Also check the response time, the behavior of the central unit when processing a large amount of data received from the connected sensors. The first scenario is shown in Figure 3 where the devices were developed and used in an experimental environment that allows monitoring the water and energy consumption of real appliances. The simulation creates benefits for the user as a convenience to access the consumption history of any smart device to create consumption strategy. The communication between the devices, central unit, and the interface is possible through a protocol created by the architecture, with this, the messages can be sent by the devices and received by the middleware. The test environment consists of an aquarium with water, a small aquarium pump, a hydrometer and a power meter. The pump, located inside the aquarium, pumps water to the hydrometer through a small pipe. The pump is plugged in to allow you to measure your energy consumption. All the equipment stays connected 24 hours a day, sending the data captured during all this time to the cloud.

From the data obtained, it is possible to compare the measurements with the values supplied by the aquarium pump manufacturer, verifying if the consumption obtained by the intelligent outlet is coherent with the theoretical consumption made available. As for the hydrometer, it compares the accumulated value recorded analogically with the sum of the digitally counted pulses, resulting in the same total value.

The second scenario was the creation of a set of lamps with three different types made using LEDs, they are a common on/off lamp, a lamp with variable intensity and a lamp with colored light (RGB). This test unites the user-friendly benefits of powering the lamps through a smartphone and the comfort of having a controllable local brightness. With the use of these lamps for local operation using a residential Wi-Fi network and a local central unit, there was an acceptable response time and, consequently, a good interaction with the lights.

At the end of the analysis of the two scenarios, it was verified that the central unit behaved correctly, did not lose any data received from the connected devices, provided the information to the user interface in a consistent manner and allowed a real-time interaction with connected equipment.

IV. Conclusions

The proposed architecture for home automation using Wi-Fi network and Web Service was implemented and validated experimentally with intelligent devices and a central unit implemented directly in the cloud. One point to highlight is the reliability obtained by the middleware because it does not present data loss in the processing of incoming and outgoing requests.

The choice to use Wi-Fi network communication makes the architecture compatible with common and simple residential routers because it does not require any converter device among several existing wireless network protocols, not losing performance during this conversion. Using HTTP(S) following the RESTful, WebSocket, and MQTT standard makes this application universally accessible, making it accessible through any user-authenticated interface, and receives data from any smart device that follows the same standard and uses the supported protocols.

the architecture does not exclude the possibility of future integration with other communication protocols, and gateways can be used to extend the range from protocols using smaller wavelengths that allow longer distance transmissions such as LoRa(long range protocol) and sigfox.

An important feature of this architecture is the possibility of being implemented in places that do not have an internet connection, that is, there is only one home network. In this type of scenario, the architecture normally works independently of the existence of the networks that allow external access. This property is also important when you have an internet connection that has oscillations or is unavailable for an indefinite period, meanwhile, the devices are communicating normally with the connected unit in place.

References

[1] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," *Big Data*, vol. 1, no. 1, pp. 51–59, 2013.

[2] B. P. Santos, L. A. Silva, C. S. Celes, J. B. Borges, B. S. P. Neto, M. A. M. Vieira, L. F. M. Vieira, O. N. Goussevskaia, and A. A. Loureiro, "Internet das coisas: da teoria à prática."

[3] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497 – 1516, 2012.

[4] M. Belshe, M. Thomson, and R. Peon, "Hypertext transfer protocol version 2 (http/2)," 2015.

[5] E. Rescorla, "Http over tls," 2000.

[6] S. Tilkov and S. Vinoski, "Node. js: Using javascript to build high-performance network programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.

[7] L. Richardson and S. Ruby, *RESTful web services*. " O'Reilly Media, Inc.", 2008.

[8] V. C. O. de Souza and M. V. C. dos Santos, "Maturing, consolidation and performance of nosql databases-comparative study," in *Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective-Volume 1*. Brazilian Computer Society, 2015, p. 32.

[9] N. Kolban, "Kolban's book on esp8266," *an introductory book on ESP8266*, 2015.

[10] A. G. W. Roeck, L. W. d. Silva *et al.*, "Smart gardens-automação e monitoramento inteligente da agricultura," 2016.

[11] V. Vassoler, "Mudança de tecnologia e como isso pode ajudar na redução de consumo na iluminação das empresas," *Revista Gestão & Sustentabilidade Ambiental*, vol. 5, no. 2, pp. 382–397, 2016.

[12] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[13] J.-S. Leu, C.-F. Chen, and K.-C. Hsu, "Improving heterogeneous soa-based iot message stability by shortest processing time scheduling," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 575–585, 2014.

[14] R. Girau, M. Nitti, and L. Atzori, "Implementation of an experimental platform for the social internet of things," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, July 2013, pp. 500–505.