A Generic Data Management Framework for the Internet of Things

Diego Fernando Núñez-Sánchez, Juan Sebastián Cantor, and Ixent Galpin ✉

Dpto. de Ingeniería, Universidad Jorge Tadeo Lozano,
Bogotá, Colombia.
{diegof.nunezs,juan.cantor,ixent}@utadeo.edu.co

Abstract. By 2025, it is estimated that there will be over 75 billion devices connected to the Internet, and it is expected that data volumes will continue to grow exponentially. As such, there will be an increasing need for effective data management of data generated by sensors in the Internet of Things (IoT). In this paper, we propose a lightweight, generic data processing framework that may be easily customized for a diverse range of data-handling IoT applications. The sensor types required by an application are specified as metadata. Sensed data is persistently stored in a horizontally partitioned relational database across IoT nodes. Subsequently, it may be retrieved by posing SQL queries over the nodes, which may be historical or real-time. Alternatively, the sensed data may also be retrieved as a JSON document over a RESTful interface, thus potentially enabling easier integration with cloud and mobile platforms. We show an instantiation of the framework for applications in the climate monitoring and security domains.

Keywords: Distributed Data Management · Query Processing · Internet of Things.

1 Introduction

The number of devices connected to the Internet is expected to grow exponentially over the next few years. Indeed, according to some estimates, by the year 2025, there will be over 75 billion "things" connected to the Internet [16], many times the predicted world population.The phenomenon is often referred to as the *Internet of Things* (IoT) [2]. IoT applications are diverse, ranging from aquatic animal tracking [6] to oil refineries [7]. However, most applications involve devices endowed with one or more *sensors*, which collect data from the surrounding environment, and *actuators*, which change aspects of the surrounding environment. Typically, IoT devices interact with applications in the cloud, and/or hardware such as smartphones, tablets or PCs. Given the data processing needs envisioned for a broad range of IoT applications, the challenge that we address in this paper is that of proposing a generic data processing framework that may be easily customized for a diverse range of data-handling IoT applications.

2019 ICAI Workshops, pp. 38–51, 2019.

This is an important problem to address given the huge economic impact that the IoT is expected to have in coming years [14]. As such, given the plethora of IoT applications likely to appear, software development costs are likely to become a greater concern. One way to reduce software development costs is by the use of generic frameworks which may be used for different applications by configuring them accordingly.

Previous solutions to provide generic processing frameworks for IoT tend to be limited. For example, IFTTT[1] has limited expressibility with regards to the data processing tasks that may be specified. Other approaches that have greater expressibility are limited to the Wireless Sensor Network domain. For example, the SNEE query processor [4] enables users to express queries using an expressive continuous query language, but have not been adapted for the IoT domain. As such, nodes in SNEE are unable to interact directly with applications in the clouds or devices such as smartphones.

This paper is structured as follows. Section 2 presents related work. A hardware design for a generic IoT node is described in Section 3, and the corresponding software architecture in Section 4. We present the middleware that we propose in Section 5. Section 6 showcases our data processing framework with an example deployment and application. Finally, Section 7 concludes.

2 Related Work

In the area of technological tools such as technologies that are responsible for making everyday tasks easier such as Paw Scout [11] which sells a chip to do the tracking of pets, among the services of the company is that the identifying labels manage to connect to the smartphones of the owners with which a social network of animals is generated and in case of loss a search network can be made in a very easy way.

Another application designed for the development of the tasks of daily life is Amazon Go [5] in which you can make purchases only with the use of a smartphone, under certain conditions in the store such as the amount of users of the same, in order not to collapse the payment system.

In the field of sensors, the company Libelium [1] is responsible for the production of sensors and provides technological solutions specific to each of the scenarios that are requested, highly reliable and with many features depending of the needs, among which are projects in which the developers implement sensors and communication systems such as the LoRa [3] to make a connection between proximity sensors and a private network LoRa private city for create a parking system around the busiest areas of the city to facilitate access to free parking in the city of Montpellier (France). In Iran, with sensors and software also provided by Libelium, a monitoring network of components in the water was generated to control them and to optimize the production processes of the fish.

[1] https://ifttt.com/

In the Internet of Things there are diverse aspects that may be considered; in particular, this work does not focus on security or scalability, since the main development of the work, will be to make available a general cluster that will be responsible for accepting any type of sensor or actuator and handle them with the benefits of using a database-based system. As proof of concept, the decision was made to use a house and three monitoring points to be able to correctly populate the information and demonstrate that the entire system that is being planned to be assembled will work correctly.

According to Sethi *et al.* [15], we can see that the security layer is not being considered for development while the others are fully covered for a development framed in the Internet of things.

Taking into account some of the data flow structures surveyed by Razzaque *et al.* [13] we can understand what is the correct order to give this data and how objects of different specifications and functionalities can all be gathered in a middleware for its correct operation and use of the data collected by the physical layer.

Jensen *et al.* [8] show the benefits of using the time series data model to reduce the amount of data that you want to be collected by the middleware sensors, but as the idea is to generate applications for in the most general cases, it was found that there are integrations that cannot be carried out in the most efficient way. This was the main reason for opting for a relational database that is also more comprehensive than all possible future works based on the structure in this document proposed and executed.

3 Hardware Design

In the stage for hardware architecture planning, some applications are already taken into account in the market and it is a matter of covering a good number of these applications with a general design that could solve the need for the other applications with a few minor modifications [18].

In the most general design of the application we want to cover the broader aspects, starting with the Figure (Right side Fig. 1) by the simplest components that are going to be responsible for doing any type of reading, where the data will be stored and the IoT area can be ordered, which is going to take care of the fact that everything is interconnected, together with the neuralgic area of the design that is in charge of processing all the data and providing the user with something readable and with added value (Left side Fig. 1).

In the area of IoT you want to have complete access to the components by means of the platform that best allows it like Raspberry Pi [17] or Arduino [9] and thus be able to have a good behavior between the elements of measurement and the way to store them, when you want to do a query generator is needed that each of the data is properly classified and to achieve this a database is implemented that meets the requirements of the system in question, communication to the central data or processing center must be done most efficiently and easily. That is why we describe a communication through a Local Area Network (LAN), in

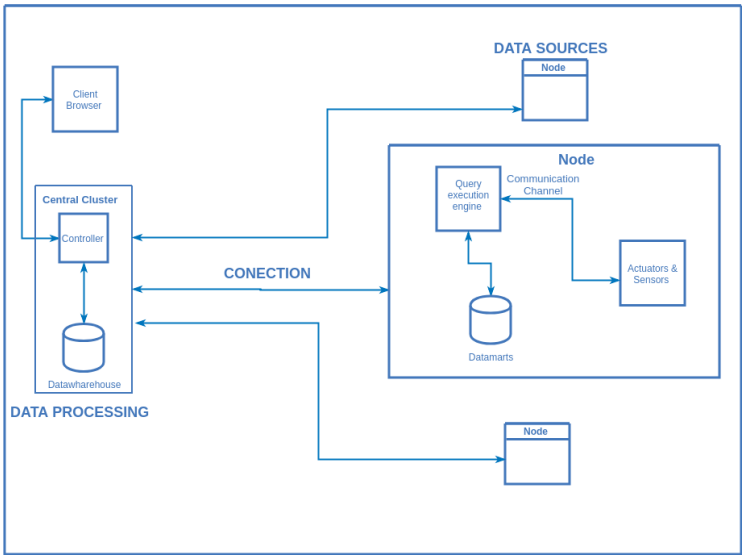


Fig. 1: Abstract hardware design, showing the main architecture.

this case, the nodes do not need to have communication between them since the application is designed to have the independence between them and thus be able to omit the same [12].

Regarding the node, the most general view is a platform with access to the internet, in which both the reading algorithms of the sensors and the logic can be executed in order to make decisions based on the data collected by the sensors and make the actuators have their effect. Along with the ability to generate a database to store each of the sensing data correctly (see Fig. 2).

In the data center or the Warehouse [10] the main characteristic is that each one of the software components that are going to be used in the nodes, or points dedicated to the IoT have in their software design the necessary elements to be perfectly compatible and associable, it is also taken into account that the consumption of the same centralized data can be given from the central data, or from a browser located on the same LAN, in order to provide all the information without any limitations of platforms [19].

To carry out the implementation of the design, it was decided to use the RaspBerry Pi platform in the nodes, due to its robustness and ease of work since it is based on Linux and its complementation with digital and analog readings through the connection pins with the which is already equipped, the set of sensors that were used to see different types of reading were binary data sensors such as the LM393 light sensor and the PIR HC-SR501 infrared motion sensor, for sensitive data this sensor is Humidity and temperature DHT11. The idea of representing all the connections of the physical layer exposed in Fig. 3 is to be able to explain the functionality that has been implemented in the operation

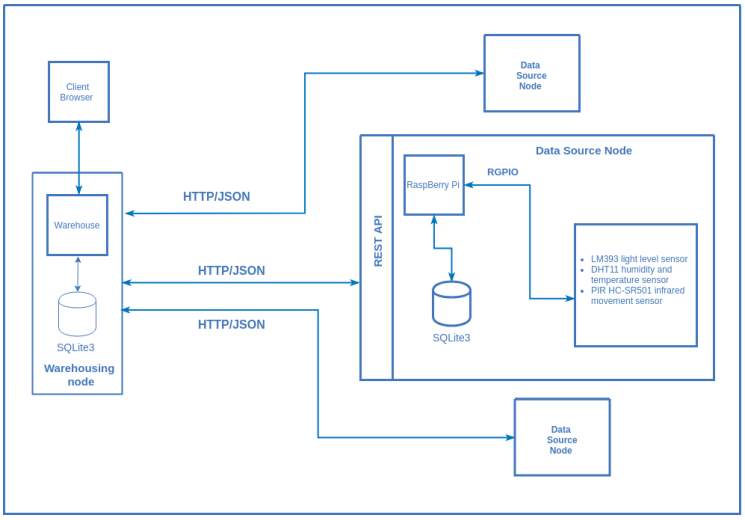


Fig. 2: Specific architecture hardware design, applied to the proof of concept

of the middleware since based on these physical connections it they have to populate the tables of the local database of each of the nodes corresponding to the metadata.

At the same time there is an application server in each of the nodes which through the LAN connection is able to publish the data to make them readable, make them available to the central data, which is responsible for centralizing the information and make it accessible to the user can enter to consume it from any platform with internet access.

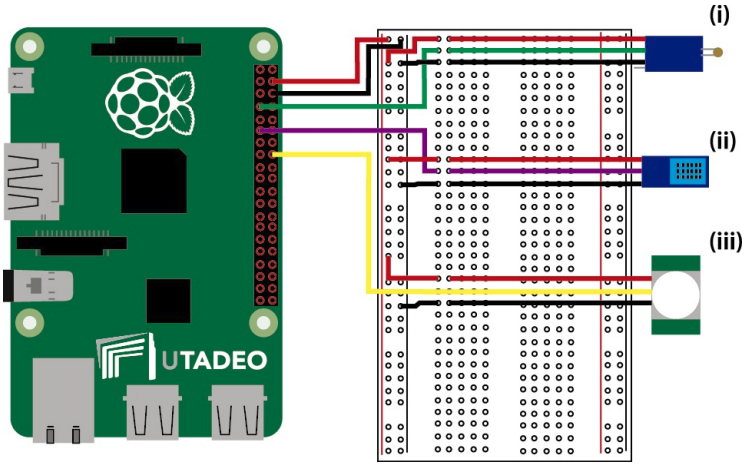


Fig. 3: Node hardware design, showing sensors connected to Raspberry Pi B+.

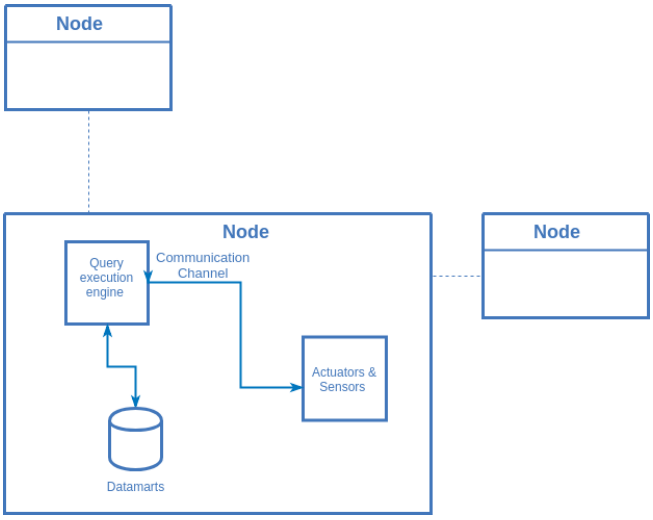


Fig. 4: Abstract architecture of a node

To make the connection of the nodes it is understood why the Raspberry Pi are so viable because they can receive several lines of information and at the same time to send information to put the actuators into operation. For the particular case of simulating an intelligent building, the connections were made as shown

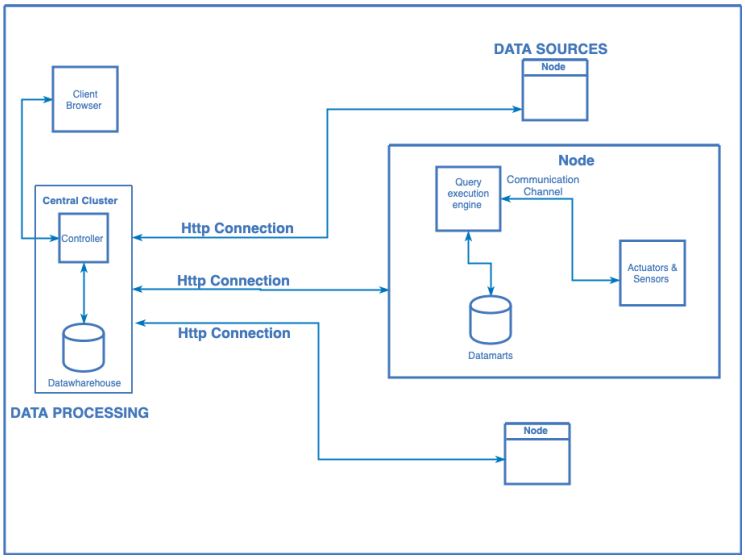


Fig. 5: Abstract design, showing the main aspects of the framework

in Fig. 3, which connects with the applications through metadata tables in the database.

Fig. 3 presents the hardware design. The sensors connected for the prototype were (i) a LM393 light level sensor connected to GPIO.IN4 (ii) DHT11 humidity and temperature sensor connected to GPIO.IN17 and (iii) a PIR HC-SR501 infrared movement sensor connected to GPIO.IN23.

4 Software Architecture

The construction of software components defines its main structure in each IoT node, the synchronized data reading sequences of the connected hardware are based on interactions controlled by time cycles using metadata parameters, the information collected is articulated concerning the node and then stored in a generic data repository.

An information web service component is available in each node to expose functional data. Each component articulates two parts: one, an architecture application server based on HTTP 2.0 and another, a software service built with Rest JSON service specifications and based on stored central data.

The data service is available for consumption, both generically by various types of software, since it uses software standard based on REST API services, and specifically, in this case, through the implementation of a solution called Warehouse Node, to define the data centralization of each connected node in the architecture.

When developing a Warehouse architecture specification, as shown in Fig. 4 and Fig. 5, the data is centralized in a software design that consolidates the

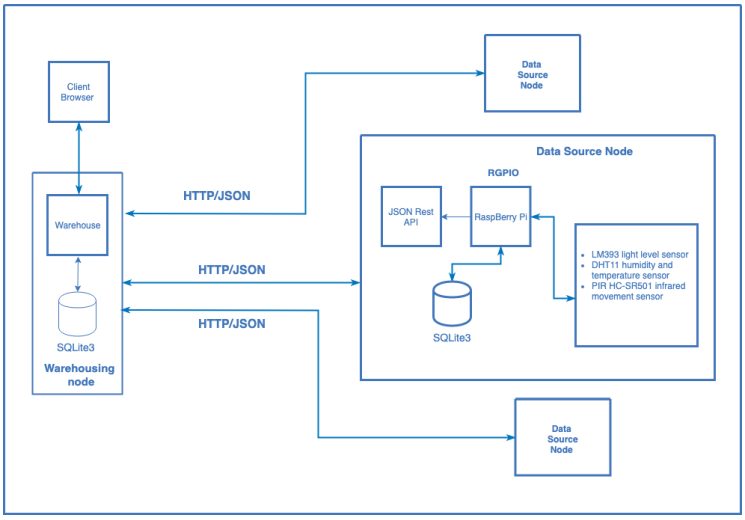


Fig. 6: Specific architecture design applied to the proof of concept

information for interpretation and analysis. So that actions can be taken on established segments of utility such as security or climate since it is part of the importance of the sensors established from the beginning of the architecture.

In order to elaborate the previous described architecture, in the first instance the construction of a project in Python 3.5 is done, it starts by reading metadata data and it is articulated with the data capture of the connected sensors depending on the reading cycle time; then, each data reading is structured, including the sensing time, for storage in SQLite 3.0 database.

The operating system is prepared with a virtual Python environment to control the necessary components, including those of Web and Rest JSON architectures, the application server used is Flask, Twilio is used for the structure of Rest JSON entities. Then, the software code prepares the information to return as a key entity value the data stored in SQLite (see Fig. 6).

The REST structures enabled in each node are used by a Spring Core specification, and Spring Boot, to centralize the data; then, through specific domain views, data is established as climate and Security. Thus, once consolidated data can be obtained, the information is articulated with a software component called Morris to graph the information. The source code used for the implementation of the prototype is freely downloadable from our Github respository[2].

5 Middleware

The implementation of Middleware for the proposed architecture consists of four hierarchically ordered elements: the first, a generic software segment in charge of obtaining raw data modeled as a key-value; the second, a data modeler that structures data for repository storage called data warehouse; the third, domain-specific views that give value to the consolidated information and, finally, the obtaining of data for analysis and decision making by means of graphs.

To obtain the data of each node, a property or configuration file is established first to articulate the IP addresses of each node included in the architecture; then, an information reading of the nodes is made through the Rest service, available with the data that is captured and, finally, articulated with abstract software entities to structure and, again, stored in a central information repository.

Within the data modeling, the value of the information obtained is determined and structured with additional repository data, where the value of the information can be determined and the additional attributes that include node identifier and sensor locations.

In the specific domain views, the data whose main value represents the usability of the sensors, the contrast with the location environment and the quality values to compare with generic values such as temperature and humidity are determined; the latter, depart from a central location of the node and can be compared with the temperature and humidity values of a city.

At the moment of graphing the information, it is consolidated by hours for the control of the data and the established reading dates, including the relevance

[2] `https://github.com/diegofnunezs/CoEduUtadeoIsTesisClusterIot`

of the spatial locations of the nodes; with this, a definition for the presentation of data determined in dot, bar or pie charts is achieved.

Technically the warehousing node is established in the following way: first, construction of code and parameterization files for reading data in the Rest specification; second, to define structures like generic software entities that allow the consolidation of information in a data container called warehouse; third, consolidated views of specific domain that include centralized information, grouped by hour and including the location of each node and, finally, the preparation of SQL queries that group and transform the information to be visualized with graphing processes.

The warehouse database has main definitions called metadata, these are basic configurations for the generic use of the proposed architecture. Within the proposed configurations, there are data read values for each node, sensors that each node can contain, and additional, connectivity configurations of hardware components where the most important is the support of heterogeneous sensors.

6 Evaluation

During the general planning of an intelligent building which is able to maintain some variables of its environment controlled and monitored, in addition to keeping them centralized in the data warehouse, together with the nodes distributed

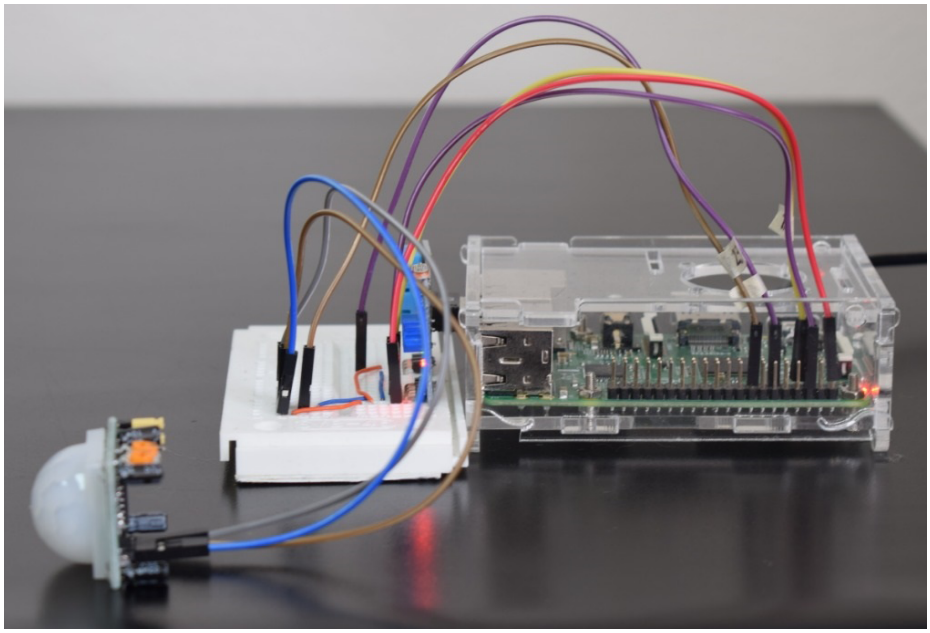


Fig. 7: Photograph showing sensors used for the proof of concept, located in the room of the house used for the test.

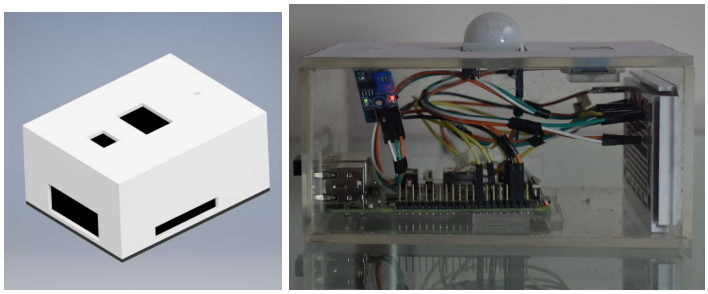


Fig. 8: This is the box that was designed for the sensor.

and connected between them to the warehouse through the LAN connection, to test each of the components described above, a test area was established for the case study.

The case study is established to monitor different locations of a house based on Internet technology of things; thus, we define three main locations in a household: the first, a room with movement, light, temperature and humidity sensors; the second, kitchen area with temperature and humidity sensors and, finally, a node located in the studio with all the architecture sensors. All the nodes

Table 1: Example queries

Query	Description
`SELECT location as LOCATION, ROUND(MAX(temperature), 2) as TEMPERATURE FROM general_view WHERE movement = 1 GROUP BY location`	Maximum temperature of the places where movement has been recorded
`SELECT location as LOCATION, time as TIME, ROUND(humidity, 2) as HUMIDITY FROM general_view WHERE temperature between 20 and 30 and light = 1`	The daily humidity of the locations with temperature between 20 and 30 degrees, which registered light
`SELECT Location as LOCATION, time as DATE FROM general_view WHERE strftime('%H:00:00',time) between strftime('%H:00:00', '00:00:00') and strftime('%H:00:00', '05:00:00') AND movement = 1 and light = 0`	Places where movement is detected and is in the night time range
`SELECT strftime('%Y-%m-%d',time) as DATE, ROUND(avg(temperature), 2) AS AVG_TEMPERATURE FROM general_view WHERE strftime('%H:00:00',time) between strftime('%H:00:00', '07:00:00') and strftime('%H:00:00', '17:00:00') GROUP BY strftime('%Y-%m-%d',time)`	Average temperature every day, between 7 a.m. and 5 p.m.

were synchronized for reading in equal times and stored in local databases with generic data structures, which will be concentrated in a single database.

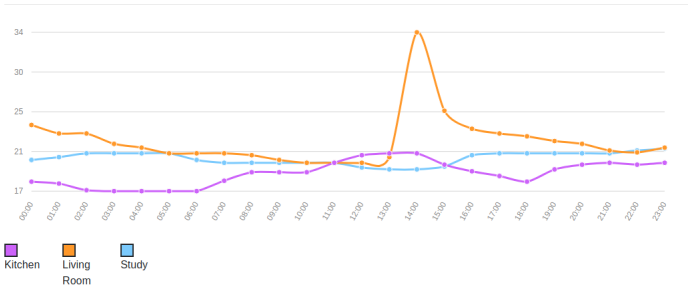
For a better organization of the sensors, it was decided to design an acrylic box with the Autodesk Inventor tool, which can be seen in the Fig. 8, which already implemented can be seen in the same image, with the design it was possible to adjust each one of the sensors in the upper part of the box to be able to make the measurements which require a wide and clear area so as not to interrupt with the correct readings of the sensors.

In Fig. 7 we can see the node that was in charge of the measurements from the study in the proof of concept.

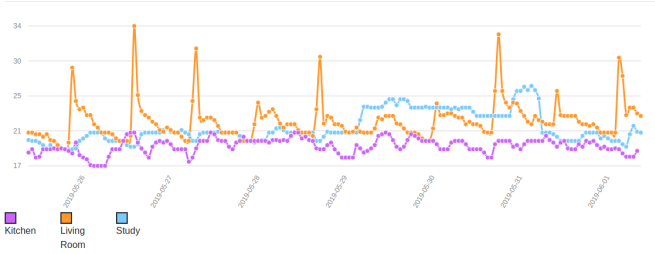
As you can see in the Fig. 8 on the right is the 3D model of the design made in Autodesk Inventor and on the right is the execution of it made in acrylic, which was very useful for its transport and subsequent commissioning.

In order to make a correct test of the sensors in the house a week of measurements was taken, during this time the nodes with 4 sensors made a writing in the database of 4036 rows and in the node of the kitchen that only had 2 of the sensors, had at the end of the 2018 week records. With all the records and using all of them there are a few graphs so that the behavior of the sensors in the test area can be observed, the graphs are divided in temperature of 1 and 7 days, humidity 1 and 7 days and a graph that represents the number of times the sensor had a positive or negative response. In this way, with the graphs we can see, in detail, the variation of the data such as humidity and temperature; in addition, there is the Table **??**, which is a list of information queries where criteria can be applied to link the various types of data. In them, you can group and delimit both with date attributes such as location and time and by specific domain values: movement, light, humidity, and temperature.

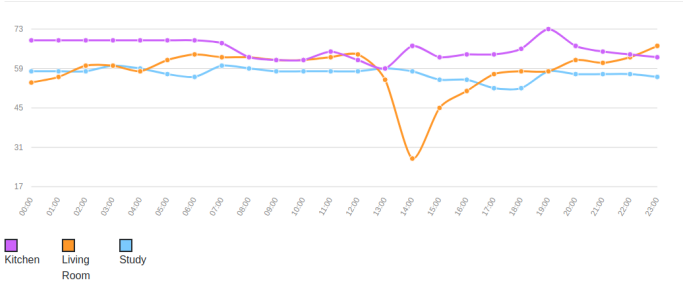
In Fig. 9a the temperatures corresponding to the day May 26, 2019, can be observed where it is observed that in the whole house an average temperature of 21° Celsius was obtained, except for the living room around 13:00 and 15:00 of the day, when a temperature rises to 34° Celsius. Fig. 9b shows the temperatures for the week of May 25 from 9:00 until 1st of June at 9:00 in 2019 where it is observed that a temperature was obtained throughout the house average of 21° degrees during the first five days, and towards the end of the week a considerable increase of the average was obtained up to 24° degrees in the living room and the study, while in the kitchen the average temperature remained of 21° degrees all week. In Fig. 9c is noted as moisture was also quite stable except for measurements made between 13:00 and 15:00 corresponding to the high temperature at the time recorded. The humidity of the week recorded and observed in the Fig. 9d is quite stable throughout the week in the kitchen being around 60% humidity, as for the living room a fall happens every day around of the noon that corresponds to the temperature rise of the same and in the study a slight decrease is observed throughout the week., it is also observed that every day around midday the living room always had a strange increase in unsubstantiated temperature



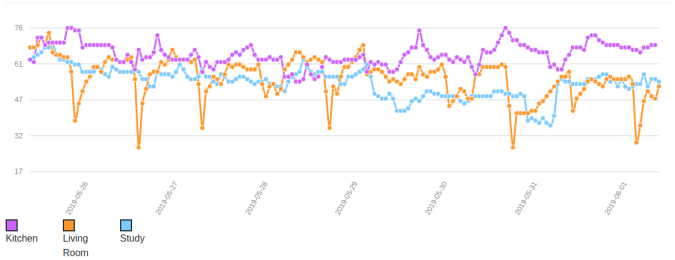
(a) One-day of test observed from Temperature sensor.



(b) Seven-day test observed from the Temperature sensor.



(c) One-day test observed from the Humidity sensor.



(d) Seven-day test observed from the Humidity sensor.

Fig. 9: Data collected during deployment trials.

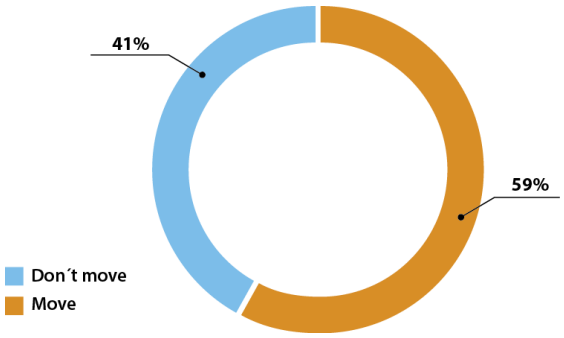


Fig. 10: Graph in which the result of seven-day test is observed from the Movement sensor in the living room.

In the Fig. 10 which represents the number of times that movement was recorded during the week in the living room, which is a social area in the house where it is seen that there were more movements than stillness.

7 Conclusions

In this paper, we have proposed a generic data management framework for the Internet of Things. We describe a generic hardware architecture for an IoT node and specify a concrete example of this implementation using the Raspberry Pi, a commonly used platform for prototyping IoT applications. We then describe a generic software architecture, in which queries can be posed using the SQLite database engine, via a RESTful interface. We envisage that such a proposal may be used for diverse IoT applications, and demonstrate an example three-node deployment in a smart home setting as a proof of concept.

Future work could usefully investigate the incorporation of semi-structured and unstructured data into the data management framework, as well as streaming multimedia data.

References

1. Asin, A., Gascón, D.: Libelium. `https://http://www.libelium.com/` (2006), accessed: 2018-05-18
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks **54**(15), 2787–2805 (2010)
3. Augustin, A., Yi, J., Clausen, T., Townsley, W.: A study of lora: Long range & low power networks for the internet of things. Sensors **16**(9), 1466 (2016)
4. Galpin, I., Brenninkmeijer, C.Y., Gray, A.J., Jabeen, F., Fernandes, A.A., Paton, N.W.: Snee: a query processor for wireless sensor networks. Distributed and Parallel Databases **29**(1-2), 31–85 (2011)
5. Inc., A.: Amazon go. `https://www.amazon.com/b?node=16008589011/` (2016)

6. Iotforall.com: Iot applications in aquatic animal tracking. https://www.iotforall.com/iot-applications-aquatic-animal-tracking/, accessed: 2019-04-21
7. Iotforall.com: Iot applications in the oil and gas industry. https://www.iotforall.com/iot-applications-oil-and-gas-industry/, accessed: 2019-04-21
8. Jensen, S.K., Pedersen, T.B., Thomsen, C.: Time series management systems: A survey. IEEE Transactions on Knowledge and Data Engineering **29**(11), 2581–2600 (2017)
9. Massimo Banzi, David Cuartielles, T.I.G.M.D.M.: Arduino. https://www.arduino.cc/ (2005), accessed: 2019-04-21
10. Paim, F.R.S., de Castro, J.F.B.: Dwarf: An approach for requirements definition and management of data warehouse systems. In: Proceedings. 11th IEEE International Requirements Engineering Conference, 2003. pp. 75–84. IEEE (2003)
11. Pawscout.com: Paw scout. https://www.pawscout.com/ (2014)
12. Perumal, V.S.A., Baskaran, K., Rai, S.K.: Implementation of effective and low-cost building monitoring system (bms) using raspberry pi. Energy Procedia **143**, 179–185 (2017)
13. Razzaque, M.A., Milojevic-Jevric, M., Palade, A., Clarke, S.: Middleware for internet of things: a survey. IEEE Internet of things journal **3**(1), 70–95 (2015)
14. Saidu, C.I., Usman, A.S., Ogedebe, P.: Internet of things: impact on economy. Journal of Advances in Mathematics and Computer Science pp. 241–251 (2015)
15. Sethi, P., Sarangi, S.R.: Internet of things: architectures, protocols, and applications. Journal of Electrical and Computer Engineering **2017** (2017)
16. Statista.com: Internet of things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions). https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, accessed: 2019-04-21
17. Upton, E.C.: Raspberry pi foundation. https://www.raspberrypi.org/ (2006), accessed: 2019-04-21
18. Vujović, V., Maksimović, M.: Raspberry pi as a sensor web node for home automation. Computers & Electrical Engineering **44**, 153–171 (2015)
19. Zorzi, M., Gluhak, A., Lange, S., Bassi, A.: From today's intranet of things to a future internet of things: a wireless-and mobility-related view. IEEE Wireless communications **17**(6), 44–51 (2010)