IoT smart spaces applied to companies' offices

Master in Computer Engineering / Mobile Computing

Javier Portaluppi

Leiria, September of 2021

Master in Computer Engineering / Mobile Computing

IoT smart spaces applied to companies' offices

Javier Portaluppi

Project report under the supervision of

Professor António Manuel de Jesus Pereira

Professor Nuno Costa

Leiria, September of 2021

Originality and Copyright

This project report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

Dedication

This work is dedicated to Sophie, my partner, my motivation, my engine, the love of my life, who pushes me to do better and be a better person each day and has encouraged me to work in my master even in the most difficult moments. And to my family back in Salta, who, regardless the reason, is always there supporting me, even the long distance and my crazy ideas.

I also want to thank all the professors involved in the master of Computer Engineering and Mobile Computing of Polytechnic Institute of Leiria who have shared their knowledge with such altruism, and David from iSolutions who passionately has given me the tools and the environment to work in this project.

This work would not be the same without any of you. Thank you all for not letting me to give up and teaching me that dreams come true only when you hardly pursue them.

Abstract

Human resource is one of the most desired and precious resource of companies nowadays. Most of company directors are totally aware of that. Keeping the right people and attract new talent is a big challenge in a world where everything moves at a fast pace.

As humans we have always tried to improve and tended to evolve. We work hard to improve our quality of life. Last years, we noticed a raise on the use of IoT and automation, to simplify process, to spend less time doing repetitive work and have more time for more important things.

This work is part of the desire of using current tools available in the market along with software development skills to aim to solve that problem, helping companies' managers to provide better places to work for their employees, which might stay longer in the company, do their work happier, and therefore improve their productivity, while being aware of the good impact on the planet when we optimize our resources.

A combined and interesting set of different languages, technologies, devices and cloud services will take place to develop an IoT ecosystem applied to workspaces which will help sensing and measuring different values inside the office to further take some actions based on them, converting, in that way, workplaces to smart places.

To carry out a work which involves different IoT devices in a cloud is still a real challenge and takes a lot of effort, but throughout this work readers will see how nowadays tools can help to solve those problems, to build a user-friendly, scalable, and multi-tenant service on the cloud, accessed by a native mobile application, meaning available for everyone at any time.

Keywords: IoT, Cloud Computing, Minew, Azure, React Native

Contents

List of Figures

List of Abbreviations and Acronyms

AI	Artificial Intelligence
AMQP	Advanced Messaging Queuing Protocol
APN	Apple Push Notification Service
BLE	Bluetooth Low Energy
CO2	Carbon dioxide
DHCP	Dynamic Host Configuration Protocol
FCM	Firebase Cloud Messaging
GATT	Generic Attribute Profile
GCM	Google Cloud Messaging
GHz	Gigahertz
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
LPWAN	Low Power Wide Area Network
MAC	Media Access Control
MQTT	MQ Telemetry Transport
PoC	Proof of Concept
RSSI	Received Signal Strength Indicator
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLM	Telemetry Data
TLS	Transport Layer Security
UID	Unique Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universally Unique Identifier

1 Introduction

Since several years, companies have hired employees to carry out different types of work so they can accomplish their mission. Depending on the sort of job, workers are provided with a set of tools to perform the best way possible. Apart from those tools, a common workspace is generally provided by the organisation for workers to do their job there, what today it is commonly known as “the office”. This is a way to share tools with everyone within the company, control the employees, ease the communication between them, among other benefits. Workers spend a considerable amount of time in that space. A common workload of 40 hours weekly results on about 1764 hours per year (of a total of 8760 hours), meaning the 20% of a common worker life, which is not a small number at all.

It is companies’ managers responsibility to provide not only the tools but a good and healthy place to work for their employees. By law, offices must respect several rules. In addition, workers may be able to perform better and longer (Ashraf A. Shikdar, February 1995). In addition, it is demonstrated that when a worker is happy in the location where he or she performs his or her job it increases the productivity (Bellet, (October 14, 2019)).

Nowadays, it is not only about performance but also getting and keeping the right people. The current market, especially for high demanded professions, as it is the IT field, is very competitive and a worker can choose a company not only because the challenge or the salary but based on benefits that the company offers to reach a great work experience. That do not exclude the workspace.

“Is it well located?” “Does it have good lightning?” “Does it have air conditioning/heater/climatize?” “Is the air fresh?” “Do they respect social distance?” “Is it noisy?” These are questions that anybody may wonder before accepting a job offer.

Latest technologies provide a set of new tools and new ways to solve problems. There are devices, which are currently offered in the market, that can be used to control, monitor and act. Temperature and humidity sensors are one of those. Keeping a good and constant temperature in the workspace, for example, is an important aspect for having a good and healthy workspace. If workers do their work in very low or hot temperatures they may not perform as expected, and it can also prevent them to get sick, resulting in a rising workload and therefore a better productivity. By measuring and monitoring temperature and humidity, it is possible to take decisions of increasing and decreasing temperature. Anyway, this is just the beginning of the flow. It will be even better if these devices communicate with each other in the local network and send the information to the cloud in order that it is accessible for anyone from anywhere at any time.

In other words, what is being described, is an IoT solution in the cloud. By this way, companies' manger can monitor their offices from all around the world and have insights of what is happening, what may happen and how to improve their offices.

It is not only about temperature and humidity measurements. There are a lot of variables that can be measured. Other interesting example on these pandemic times are carbon dioxide (CO2) sensors, since they can measure how pure is the air and when is time to refresh it. Luminosity sensors as well, to measure how good the light is to perform the work properly and avoid eyes disorder or other diseases. Occupancy sensors, to swich on or off ventilation when someone enters or exits an office, among many others.

On the other hand, clouds services have evolved incredibly last years, providing many innovative, useful and effortless services for common solutions that are being used more and more. Nowadays, main cloud services providers offer IoT as a platform and IoT as a service which ease the development for this type of solutions and the time to market as well. Thanks to these services and platforms, it is possible to gather and store data received from sensors. However, one part of the solution is still missing.

Once sensors send data to the cloud and data is processed and stored there, the next and final step is to show the data to final users. These can be done in several ways. For example, a web report in the cloud. Anyway, if we want to provide a better and real time experience, a mobile application should be developed. By this way, final users are able not only to consult office' information but to receive some push notifications when some pre-configured alerts are triggered.

The combination and smart usage of IoT, current cloud services and last trend in mobile development will lead to an integrated smart solution to monitor workspaces to take decision based on real time data and take actions, making the task easy for companies' manager to provide a better place to work for the employees of the company.

Having a IoT smart space for companies is not only a benefit for their employees but also for the company itself. By the usage of this type of solutions, they can manage resource more efficiently and, therefore, save energy, which finally result to a reduce of costs while making a contribution to the planet.

1.1 Objectives and goals

It is proposed within this work to create an ecosystem for companies' manager to monitor their offices without having to be physically there, providing information that can help to take decisions to improve the workspace. It will be based on sensors placed in the office, which will gather information about the environment and will send data to the cloud through a

gateway. Data will be analysed on a cloud platform and presented to the user as a report on the web. It is also part of this proposal, to create a mobile application which will show the information of the sensors of given office and receive real time notifications when determined actions happen.

This work will be carried out and applied to iSolutions Barcelona office but can be easily deployed and implemented to any office of the world with internet access.

1.2 Structure

In the first part of the work, an *List* of Figures

List of Abbreviations and Acronyms

AI	Artificial Intelligence
AMQP	Advanced Messaging Queuing Protocol
APN	Apple Push Notification Service
BLE	Bluetooth Low Energy
CO2	Carbon dioxide
DHCP	Dynamic Host Configuration Protocol
FCM	Firebase Cloud Messaging
GATT	Generic Attribute Profile
GCM	Google Cloud Messaging
GHz	Gigahertz
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
LPWAN	Low Power Wide Area Network
MAC	Media Access Control
MQTT	MQ Telemetry Transport
PoC	Proof of Concept
RSSI	Received Signal Strength Indicator
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLM	Telemetry Data
TLS	Transport Layer Security
UID	Unique Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universally Unique Identifier

Introduction to technical concepts will take place, for readers to be in the same page when it comes to the project development. In this section, *Theoretical framework & Technologies*, concepts such as Bluetooth, IoT, Cloud services and mobile application are developed.

Next section is *Related Work,* where some similar projects on the same field, which have been encouraged the research of this project, are briefly described.

Once readers are up to date with technology to be used, *Architecture* of the solution will be presented and each component of the system is described and analysed.

Next chapter developed is *Implementation*. Within it, readers will find how the architecture was put into practice, explaining in detail the technology used for each component described in the previous section and also explaining their output.

2 Theoretical framework & Technologies

2.1 Bluetooth

There are billions of Bluetooth devices in the world. We use this technology for transferring files between devices, listening to music on our headsets regardless having wires, for communicating with our car through voice commands while we are driving. It is also commonly used in smart homes, fitness equipment, health, IoT, industry 4.0, among other plenty of uses. It is easy to use, cheap, secure, its speed is reasonable and, last but not least, its last versions can work using low amount of energy, providing a lot of benefits to manufacturers and final users. Bluetooth is a wireless communication standard, a set of protocol layers and profiles, that works over short distance using short-length radio waves at 2.4GHz. Bluetooth does not need a router or access point to perform the connection between two devices. It just needs the two Bluetooth adapters and once the connection is established the data may be transferred bidirectionally. Although the communication is always between two devices, multiple devices can be connected at the same time, and it is easy for the users to switch between current connections. Its range covers a radius of about 10 meters. Bluetooth has been created to provide a wireless alternative to wired protocols. Several versions have been developed since its creation. They are described below in Figure 1. Bluetooth version through the years

Year Introduced	Bluetooth Version	Feature
2004	2.0	Enhanced Data Rate
2007	2.1	Secure Simple Pairing
2009	3.0	High Speed with 802.11 Wi-Fi Radio
2010	4.0	Low-energy protocol
2013	4.1	Indirect IoT device connection
2014	4.2	IPv6 protocol for direct internet connection
2016	5.0	4x range, 2x speed, 8x message capacity + IoT

Figure 1. Bluetooth version through the years

(Jaycon Systems, 2017)

Clearly, one of the major advances was version 4.x in 2010 with the introduction of BLE, Bluetooth Low Energy. It opened a world of possibilities, especially in IoT, fitness, health care and the industry. In addition, it promoted the creation of BLE Beacons. These are hardware devices that broadcast packets of data to nearby devices using the same

protocol. They can be powered by battery, through USB or even by using tiny solar panels, allowing their installation to be anywhere. Generally, BLE Beacons are used for advertising purposes. It means that they broadcast a packet to nearby devices which detect it and trigger an action depending on that packet. For example, let us suppose that user is in a big mall where beacons have been installed and configured, and user has downloaded the mall's app. When the user approach to a Beacon, he can get a notification on his smartphone saying which is the closest emergency exit. Trilateration can also be used to get user's current position in indoors environments. With BLE, this is not as precise as GPS but: it uses lower battery, it is cheaper and last but not least, it works in indoors where GPS is limited. This means that, in the example given, user even might be able to navigate, using Bluetooth, to his closest exit.

There are two main Beacons protocols. The first one to be created was iBeacon. It was introduced by Apple in mid-2013. iBeacon beacons can only broadcast its Universally Unique Identifier (UUID). It means that the application that receives the broadcast must know what this UUID means and trigger some event based on that; it is in developer's hand. Users must have a specific iOS application to react against iBeacon beacons broadcast. On the other hand, Google has introduced its protocol as well, Eddystone. This one, not only can broadcast a UUID (Eddystone-UID) but can broadcast a URL (Eddystone-URL), a time-varying beacon frame (Eddystone-EID) or even telemetry data (Eddystone-TLM) for instance battery voltage, beacon temperature, number of packets sent since last start up, and beacon uptime.

BLE networks include the peripheral devices (the low power nodes) and a central device (the gateway) to which the peripheral devices connect. The peripheral device can set an advertising interval, and every time this interval passes, it will retransmit its main advertising packet. Alternatively, a listening device can request the response data. Peripheral devices must advertise themselves to allow a connection to be established. The Generic Attribute Profile (GATT) establishes in detail how to exchange all profile and user data over a BLE connection (Kevin Townsend, 2021). GATT transactions allow the transfer of data from a peripheral device to a central device, using a Bluetooth stack with three high-level objects: profiles, services and characteristics. The profile defines the client/server relationship and services are used to break up data into logical entities which contain specific data called characteristics (Josie Hughes, 2015). Each service distinguishes itself from others using a unique numeric ID (UUID). The standard GATT maximum transfer unit used to be 23 bytes, three of which are taken up by the GATT protocol, leaving 20 bytes per transmission, but nowadays this limit has been extended to 512 bytes.

2.2 IoT

The concept of IoT or internet of things makes reference to a type of network based on “things”, physical objects that are embedded with sensors, software and other technologies which, using different protocols, are connected with each other and other systems through internet. The number of IoT devices worldwide in 2020 is about 8.74 billion, and is forecast to almost triple, to more than 25.4 billion, in 2030. In 2020, the highest number of IoT devices is found in China with 3.17 billion devices. (Holst, 2021)

The Internet of Things is not a single technology, but it is a mixture of different hardware & software technology.

2.2.1 Characteristics

Some of the characteristic of IoT are:

- Interconnectivity. Anything can be interconnected using the communication infrastructure.
- Things related service. IoT is capable to provide things related service.
- Heterogeneity and interoperability. Different types of devices using different protocol are able to interact within the system with other devices and networks. As it is shown in *Figure 2. Heterogeneity and interoperability in IoT*, implementing an IoT solution always mean dealing with different technologies, protocols, devices (sensors and other hardware), systems/applications and services.
- Dynamic changes. The state and number of devices changes dynamically.
- Scalability. Network and devices management are planned beforehand to ensure horizontal and vertical growing.
- Security and Safety: IoT must take into account safety, taking into account not only protecting private data but also taking into account humans physical well-being.
- Connectivity. Enabling network accessibility, getting on a network, and compatibility, ability to produce and consume data.
- Cost effective and consumption efficient. (Holst, 2021)

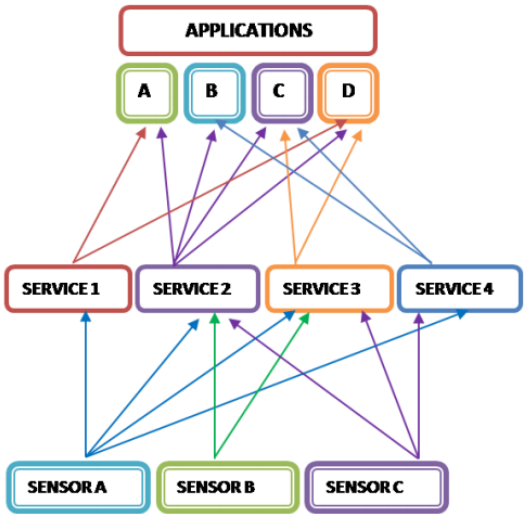


Figure 2. Heterogeneity and interoperability in IoT

(Holst, 2021)

2.2.2 Architecture

IoT architecture consist in a set of layers of technology which support it. These layers are described below and shown in Figure 3. IoT architecture

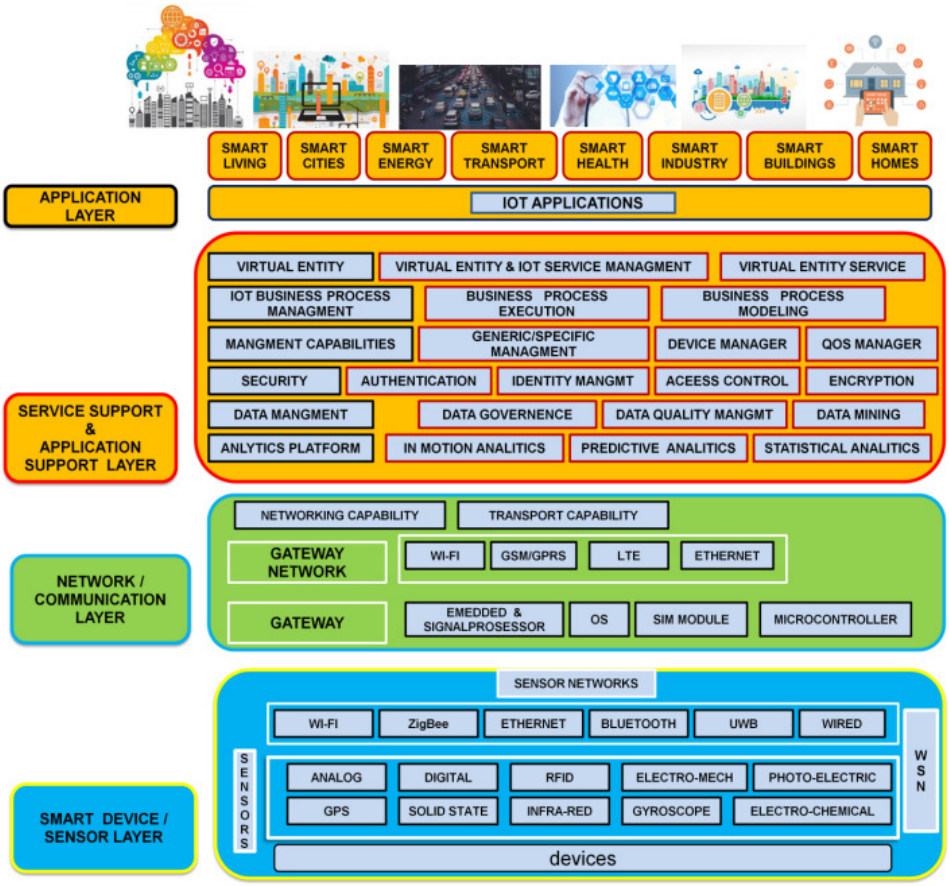


Figure 3. IoT architecture

(Keyur K Patel, 2016)

2.2.2.1 Smart device / sensor layer

This is the lowest layer, and it is based in sensors (or actuators) which allow interconnection between digital and physical signals to capture data from the environment or act on it. They implement network capabilities of different protocols such as Ethernet, Wi-Fi and Bluetooth to receive and send data to the network layer to be collected and processed, playing an important role in real time communication feature. There are different types of sensors, and they can measure temperature (see an example in Figure 4. DHT11. A common temperature and humidity sensor used in researching IoT projects), humidity, pressure, speed, air quality among others.

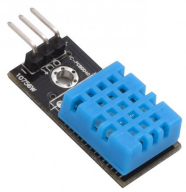


Figure 4. DHT11. A common temperature and humidity sensor used in researching IoT projects

2.2.2.2 Gateway and Networks layer

Sensor device can generate a massive quantity of data, which needs to be filtered, processed and sometimes stored. This layer communicates directly with the layer described above, making the transmission of data to the service layer possible by the use of diverse network protocols depending on the convenience. Gateways, for example, are devices which works on local area networks, gathering and filtering packages send by sensors and sending them to service layer. They can also work the other way around. It means that can send data from the service layer to a specific sensor.

2.2.2.2.1 About IoT Gateways

IoT gateways are dedicated hardware applications used to connect the devices to the network, allowing the conversion of data between the short distance communication protocols to the traditional communication network. The gateway is supposed to support different types of sensor nodes, multiple communication protocols, both wireless or wired, and provide a set of unified information for the application or user, making these only responsible for data processing. (André Glória, 2017)

An IoT Gateway is a solution for enabling IoT communication, usually device -to-device communications or device-to-cloud communications. The gateway is typically a hardware device housing application software that performs essential tasks. At its most basic level, the gateway facilitates the connections between different data sources and destinations. In Figure 5. IoT gateways, it shown how IoT gateways interact with sensors and the cloud.

IoT Gateways have evolved to perform many tasks, from simple data filtering to enabling visualization and complex analytics. These smart devices are helping power the current wave of IoT expansion.

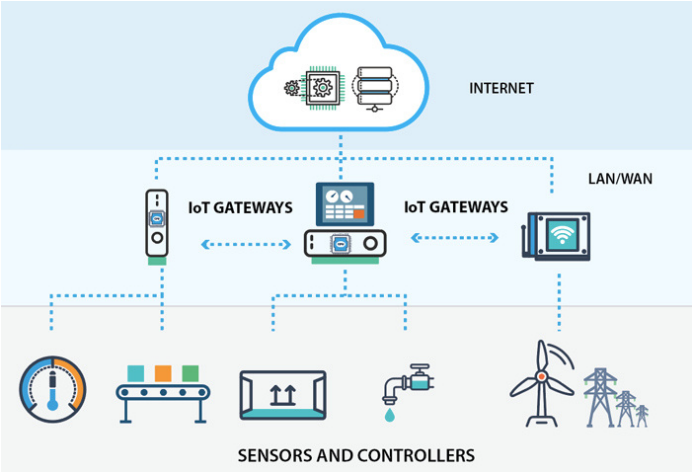


Figure 5. IoT gateways

A versatile IoT Gateway may perform any of the following functions:

- Facilitating communication with legacy or non-internet connected devices.
- Data caching, buffering and streaming.
- Data pre-processing, cleansing, filtering and optimization.
- Some data aggregations.
- Device to Device communications/M2M.
- Networking features and hosting live data.
- Data visualization and basic data analytics via IoT Gateway applications.
- Short term data historian features.
- Security – manage user access and network security features.
- Device configuration management.
- System diagnostics.

(Open Automation Software, 2021)

An example of an IoT Gateway is shown in Figure 6. LSP8 Lora WAN gateway.

Figure 6. LSP8 Lora WAN gateway

Some of the applications of a gateways are:

- iBeacon / Eddystone / other BLE tag receiver.
- BLE sensor reader / receiver.
- Asset tracking.
- Access control management.
- Advertisement promotion.
- Indoor location and position.
- Industrial automation.

2.2.2.3 Management Service layer

The management service layer process information through analytics and security controls, modelling and managing of devices and rule engines. It also provides communication with different objects and systems and executes functions of routing and filtering data and triggering different actions depending on the rule engine.

When it comes to analysing data, different tools can be used. In this task, it is important to extract information from a massive amount of raw data provided by the sensors at fast rate, also providing a quick response rate to the users of the system. In memory analytics allows large volumes of data to be cached in the random-access memory (RAM), instead of physical or virtual disk, which reduces data query time and therefore decision-making time. On the other hand, in streaming analytics, another type of analytics, data analysis is carried out on real time. Streaming analytics is the ability to constantly calculate statistical analytics while moving within the stream of data, opening the possibility to connect with external data source.

That being said, there are two data paths to receive, process and store information (Lambda Architecture for IoT and Big Data Systems, 2019):

1. **Cold path.** Data from the input stream(s) is stored and then processed at some later time. This can be beneficial when is needed to store database to create some reports in the future, use them in machine learning project and so on.
2. **Hot path.** Data from the input stream(s) is processed in near real-time. This is useful for all systems that need to process data and take some kind of action in real-time. For example, in IoT solutions. If a theft is occurring, we want the system to notify users as soon as possible and react, just to mention one of thousand cases of use.

Data management is the ability to manage data information flow. With data management in the management service layer, information can be accessed, integrated and controlled. Higher layer applications can be shielded from the need to process unnecessary data and reduce the risk of privacy disclosure of the data source. Data filtering techniques such as data anonymisation, data integration and data synchronization, are used to hide the details of the information while providing only essential information that is usable for the relevant applications. With the use of data abstraction, information can be extracted to provide a common business view of data to gain greater agility and reuse across domains. Security must be enforced across the whole dimension of the IoT architecture right from the smart object layer all the way to the application layer. Security of the system prevents system hacking and compromises by unauthorized personnel, thus reducing the possibility of risks. (Keyur K Patel, 2016)

2.2.2.4 Application layer

IoT applications covers smart environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy and many more. Next, some of them will be described.

2.2.3 Application Areas

Potential applications of the IoT are numerous and diverse, permitting into practically all areas of every-day life of individuals, enterprises, and society as a whole. The IoT application covers “smart” environments/spaces in domains such as the ones described below. (Priyadarshiny, 2020)

2.2.3.1 *Wearables*

It is a trend right now and probably the most commercialized IoT as a service. It consists in gadgets that we can wear and sense status of our body, such as heart rate or glucose. One of the most popular in the market is FitBit (Figure 7. FitBit), a smart watch to monitor heartbeat, manage stress, monitor sleep, among others features.

Figure 7. FitBit wearable

2.2.3.2 *Health care*

IoT applications can turn reactive medical-based systems into proactive wellness-based systems and opens ways to a sea of valuable data through analysis, real-time field data, and testing (Figure 8. IoT in health care). The Internet of Things also improves the current devices in power, precision, and availability. IoT focuses on creating systems rather than just equipment.

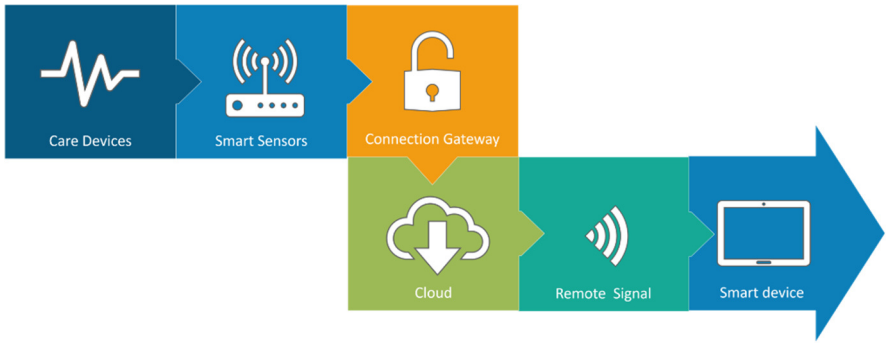


Figure 8. IoT in health care

(Priyadarshiny, 2020)

2.2.3.3 *Smart cities*

With the application of IoT in cities it is possible to monitor traffic, population behaviour and to reasonably increase security and fidelity to law. Also, analysing this type of data from cities may lead to investigations to improve the city itself and even design better cities in the future, tending to a better quality of life, better usage of space while collaborating with

the environment. For example, an IoT system in a parking lot which lead the driver to the closest park slot, not only improves driver time but also generates less wasting as the car engine is turned on less time.

2.2.3.4 Agriculture

One of the most common application of IoT in agriculture is the creation of smart greenhouses. A greenhouse farming technique enhances the yield of crops by controlling environmental parameters.

Through the implementation of IoT sensors, a significant amount of data can be obtained on the state and stages of the soil. Information such as soil moisture, level of acidity, the presence of certain nutrients, temperature, and many other chemical characteristics, helps farmers control irrigation, make water use more efficient, specify the best times to start sowing, and even discover the presence of diseases in plants and soil.

A greenhouse with embedded devices not only makes it easier to be monitored but also, enables us to control the climate inside it. Sensors measure different parameters according to the plant requirement and send it to the cloud. It, then, processes the data and applies a control action. An example of a real-word IoT greenhouse flow is distinguished in Figure 9. IoT greenhouse example.

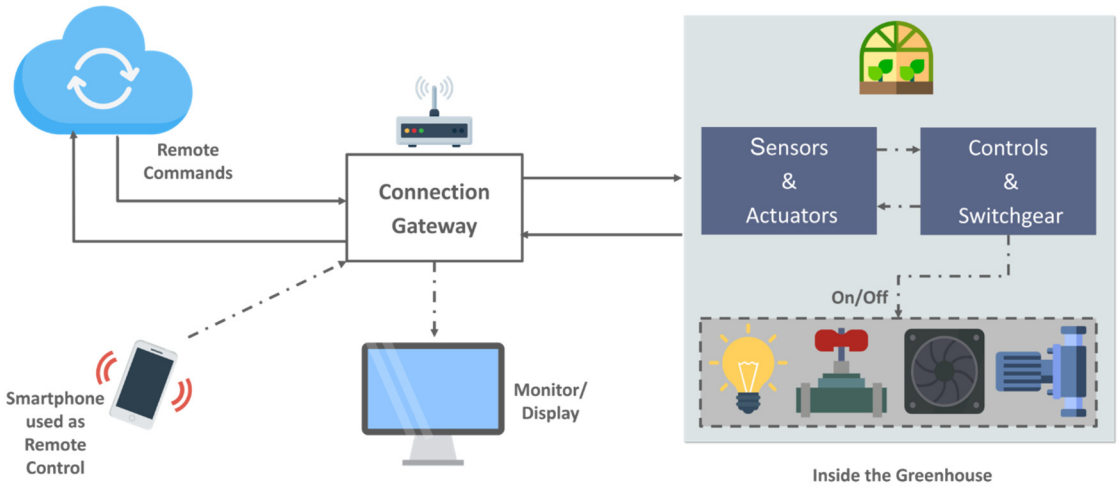


Figure 9. IoT greenhouse example

(Priyadarshiny, 2020)

2.2.3.5 Industrial automation

IoT applications in industrial environments opened the door to even re-engineer products and their packaging to deliver better performance in both cost and customer experience.

- Factory digitalization.

- Product flow monitoring.
- Inventory management.
- Safety and security.
- Quality control.
- Packaging optimization.
- Logistics and supply chain optimization.

Also, regarding security, IoT is very helpful, by detecting gas and leakage presence in industrial environments, and providing tools for early detection of equipment malfunctions and service maintenance management.

2.2.3.6 Smart spaces

When it comes to smart spaces, application of IoT drives to enhance quality of life of people in the space, automate tasks and thus increase productivity, strengthen safety, among others. Weather monitoring, air pollution monitoring, water quality and consume, identification of people inside the space, remote controlling and energy saving are some of the results of IoT applied to smart spaces.

2.2.3.6.1 Smart homes

Switching on/off devices automatically and/or remotely to save energy, smart fridges which tell the user what is missing in it prepare the next shopping list, accessing remotely to washing machine to verify status, temperature control, intrusion detection systems, water supply monitor are just some of the applications of IoT in homes.

2.2.3.6.2 Smart offices

We are entering to an era of smart office, but what does that really mean? What makes an office smart? A smart office is a workplace where a workplace where modern technology, including but not limited to IoT solutions, is leveraged to help employees work smarter, better, and faster, increasing by this way their productivity. This is achieved by removing repetitive tasks and unnecessary obstacles, allowing workers to focus on what really matters. Typically, a smart office solution consists of a suite of technologies that connect with the employees, the building and existing IT infrastructure to achieve these goals. Benefits of having a smart office solution are described below (Senion, 2019):

Improve collaboration. Smart office solutions tend to ease collaboration and organization between employees. A meetings organizer could be a clear example of this.

Boost productivity. Smart offices allow workers to put all the efforts in their jobs instead of other tasks that can be automated or made by a computer. Following the example above

mentioned, by having a meeting organizer, employees do not have to waste time looking for meeting rooms or time gaps, and harness that time to keep working.

Better use of office space. Smart office solutions can help unlock more space by optimizing the use of space. They can detect “no shows” in meeting rooms and even predict new ones. Next time, meeting can be organized in a smaller room.

Actionable utilization insights. By storing, processing and analysing insights about office status it is possible to enhance decision making to provide a better way of work, better organization of spaces and smarter work.

Attract and retain employees. Professionals today expect a lot more from their workspace than before. They are commonly used to work with latest technology and that is not the exception when they are at the office. By providing modern smart office solutions, companies can provide an employee experience that helps attract and retain employees. Having a good temperature, a good lighting, fresh and not polluted air are just some examples.

Improve the wellbeing of staff. Replacing repetitive tasks for a smart solution and providing better places to work is not only beneficial for the company, increasing employee’s productivity, but also for employees reducing level of stress. Working smarter is getting more done with less stress and less effort being that beneficial for employees and employers.

Shorter mean time to repair. By streamlining the employee-to-facility management channel, smart office solutions can radically reduce the time and effort to report broken or missing equipment. Easy and effortless reporting lead to quicker repairs.

2.3 MQTT

“MQTT is a client-server publish-subscribe messaging transport protocol. It is light weight, open, simple, and designed to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in machine-to-machine and IoT contexts where a small code footprint is required and/or network bandwidth is at a premium.

The protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bi-directional connections. Its features include:

- Use of the publish/subscribe message pattern which provides one-to-many message distribution and decoupling of applications.
- A messaging transport that is agnostic to the content of the payload.

- Three qualities of service for message delivery
  1. "At most once", where messages are delivered according to the best efforts of the operating environment. Message loss can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.
  2. "At least once", where messages are assured to arrive, but duplicates can occur.
  3. "Exactly once", where messages are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.
- A small transport overhead and protocol exchanges minimized to reduce network traffic.
- A mechanism to notify interested parties when an abnormal disconnection occurs. “ (OASIS, 2019)

MQTT clients are very small, require minimal resources so can be used on small microcontrollers. MQTT message headers are small to optimize network bandwidth. MQTT allows messaging between device to cloud and cloud to device. This makes easier the fact of broadcasting messages to groups of things. MQTT can scale to connect with millions of IoT devices. Reliability of message delivery is important for many IoT use cases. This is why MQTT has 3 defined quality of service levels: 0 - at most once, 1- at least once, 2 - exactly once. Many IoT devices connect over unreliable cellular networks. MQTT’s support for persistent sessions reduces the time to reconnect the client with the broker. MQTT makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth. (MQTT, 2021)

MQTT excels when transferring data over the wire in comparison to protocols like HTTP. MQTT is extremely easy to implement on the client side. Ease of use was a key concern in the development of MQTT and makes it a perfect fit for constrained devices with limited resources today.

In March 2019, OASIS ratified the new MQTT 5 specification. This new MQTT version introduced new features to MQTT that are required for IoT applications deployed on cloud platforms, and those that require more reliability and error handling to implement mission-critical messaging.

The MQTT protocol is the most popular and best received Internet of Things protocol in the world today (Figure 10 Interest in MQTT (light blue) compared with other protocols according google searches.).

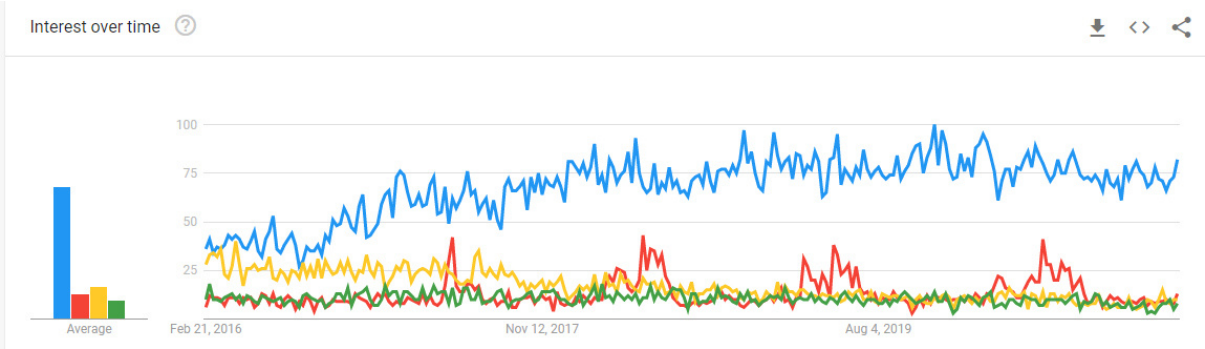


Figure 10[1] Interest in MQTT (light blue) compared with other protocols according google searches. MQTT in light blue, COAP in red, XMPP in yellow and AMQP in green.

2.3.1 The publish/subscribe pattern

The publish/subscribe pattern (also known as pub/sub) provides an alternative to traditional client-server architecture. In the client-server model, a client communicates directly with an endpoint. The pub/sub model decouples the client that sends a message (the publisher) from the client or clients that receive the messages (the subscribers). The connection between them is handled by a third component (the broker). The job of the broker is to filter all incoming messages and distribute them correctly to subscribers. A common and practical flow for publishing/subscribing to temperature topic can be visualized in Figure 11. MQTT overview

.

The most important aspect of pub/sub is the decoupling of the publisher of the message from the recipient (subscriber). This decoupling has several dimensions:

- Space decoupling: Publisher and subscriber do not need to know each other (for example, no exchange of IP address and port).
- Time decoupling: Publisher and subscriber do not need to run at the same time.
- Synchronization decoupling: Operations on both components do not need to be interrupted during publishing or receiving.

[1] https://trends.google.de/trends/explore?date=today%205-y&q=mqtt,coap,xmpp,amqp

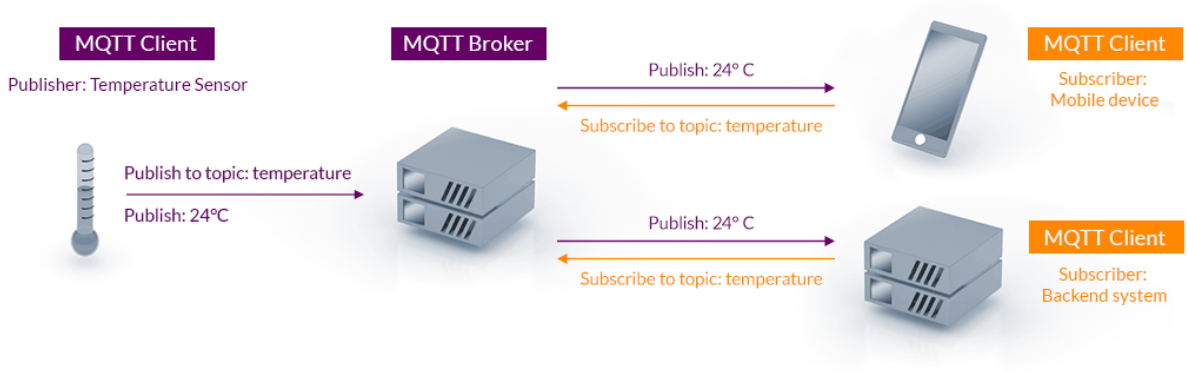


Figure 11. MQTT overview

(MQTT, 2021)

MQTT uses subject-based filtering of messages. Every message contains a topic (subject) that the broker can use to determine whether a subscribing client gets the message or not. The MQTT protocol is based on TCP/IP. Both the client and the broker need to have a TCP/IP stack.

2.3.2 Clients

MQTT clients publish a message to an MQTT broker and other MQTT clients subscribe to messages they want to receive. Both publishers and subscribers are MQTT clients. The publisher and subscriber labels refer to whether the client is currently publishing messages or subscribed to receive messages (publish and subscribe functionality can also be implemented in the same MQTT client). An MQTT client is any device (from a micro controller up to a full-fledged server) that runs an MQTT library and connects to an MQTT broker over a network. (HiveMQ, 2019)

2.3.3 Brokers

MQTT brokers receive published messages and dispatch the message to the subscribing MQTT clients. An MQTT message contains a message topic that MQTT clients subscribe to and MQTT brokers use these subscription lists for determining the MQTT clients to receive the message. The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients. The broker also holds the session data of all clients that have persistent sessions, including subscriptions and missed messages. Another responsibility of the broker is the authentication and authorization of clients. (HiveMQ, 2019)

2.4 Cloud Services & Azure

Cloud services are infrastructure, platforms, or software that are hosted by third-party providers and made available to users through the internet. All infrastructure, platforms, software, or technologies that users access through the internet without requiring additional software downloads can be considered cloud computing services.

- Infrastructure-as-a-Service (IaaS) provides users with computing, networking, and storage resources.
- Platforms-as-a-Service (PaaS) provides users with a platform on which applications can run, as well as all the IT infrastructure required for it to run.
- Software-as-a-Service (SaaS) provides users with—essentially—a cloud application, the platform on which it runs, and the platform's underlying infrastructure.
- Function-as-a-Service (FaaS), an event-driven execution model, lets developers build, run, and manage app packages as functions without maintaining the infrastructure. (RedHat, 2021)

There are diverse cloud providers that offer these services. One of those is Azure. Built by Microsoft, Azure is a cloud platform with plenty products and cloud services designed to solve problems and innovate with new ideas. One of the domains covered by Azure are IoT solutions.

Azure IoT is a collection of managed and platform services across edge and cloud that connect, monitor, and control billions of IoT assets. It also includes security and operating systems for devices and equipment, along with data and analytics that help businesses to build, deploy, and manage IoT applications.

Azure IoT solutions consider 3 components:

1. **Things**

The physical objects, or things, such as industrial equipment, devices, or sensors, that connect to the cloud persistently or intermittently. (Figure 12. Azure IoT: Things.)

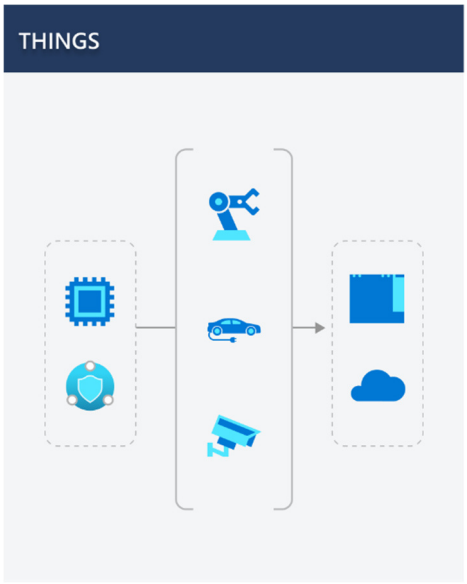


Figure 12. Azure IoT: Things.

2. Insights

Information collected by the things, which is analysed and turned into actionable knowledge either by people or AI (artificial intelligence). (Figure 13. Azure IoT. Insights)

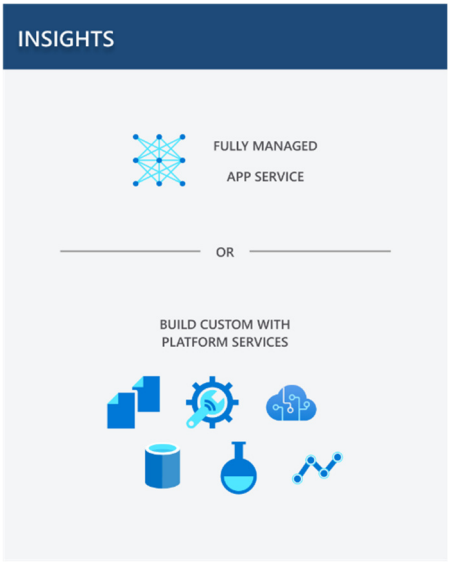


Figure 13. Azure IoT. Insights

3. Actions

The way people respond to those insights and connect them to their business, as well as the systems and tools they use. (Figure 14. Azure IoT. Actions)

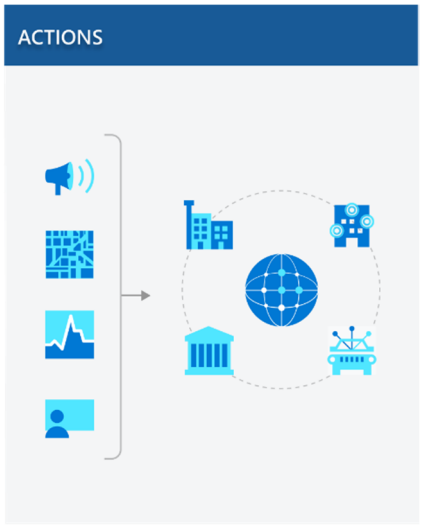


Figure 14. Azure IoT. Actions

2.4.1 Benefits of Azure IoT

- Ease development, providing simplicity and integrability.
- Secure environment, through partnership with leaders in IoT security area.
- Ready for innovation, by enhancing data, applying AI and analytics solutions.
- Performance and reliability even in extended offline periods.
- Scalability, allowing to begin small and grow rapidly as desired.

(Microsoft, 2021)

2.4.2 Azure IoT approaches

Azure offers two approaches for IoT solutions:

1. **Azure IoT Hub** [2]

Enables highly secure and reliable communication between your IoT application and the devices it manages. Azure IoT Hub provides a cloud-hosted solution back end to connect virtually any device.

Azure IoT Hub and Azure Digital Twins provide the building blocks for companies to construct customized solutions for complex IoT scenarios.

2. **Azure IoT Central** [3]

[2] *https://azure.microsoft.com/en-us/services/iot-hub/*

[3] *https://azure.microsoft.com/en-us/services/iot-central/*

Azure IoT Central is highly secure, scales with the business as it grows, ensures client's investments are repeatable, and integrates with the existing business apps.

Azure IoT Central enables companies to get started quickly building IoT applications with a fully managed IoT solution offering. (Microsoft, 2021)

Main difference between these approaches is that Azure IoT Central is a software as a service, easing the building of a new solution very quickly whereas Azure IoT Hub is a platform as a service which is more customizable at a cost of needing more expertise to start using it. Azure IoT central uses Azure IoT hub under de covers. Although it is possible to use MQTT protocol within them, they are not a full-featured MQTT broker and does not support all the behaviours specified in the MQTT v3.1.1 standard but they do enable devices to communicate with the platform using MQTT v3.1.1. All device communication with IoT Hub must be secured using TLS/SSL. When it comes to MQTT 5, its support is still in preview.

2.5 Mobile Development and Push notifications

One of the characteristics of mobile applications is the sending and receiving of messages through a system of notifications known as "PUSH" that consists of requests appearing on the display of the smartphone at a scheduled time. The content and the time to be sent are totally customizable. One of their particularities is that they are asynchronous, it means that, it is not expected or required for the recipient to answer a message. However, PUSH notifications are proactive, as they offer visual alerts to inform the recipient of a message or event received and invite them to act on them. On receiving the notification, the user can interact in different degrees, from merely reading it to answering it, thus permitting feedback.

This mechanism clearly opens the door to a fluent communication from a service to users in real-time, becoming a very interesting point for IoT solutions. By this way, users can be notified of what is happening in the eco-system (smart home, smart city, smart industry) instantly and even take actions based on that information.

The steps below explain how push notification works on android devices (Figure 15. Android push notifications explained

):

1. First Android device sends sender id, application id to GCM (Google Cloud Messaging) server for registration.
2. Upon successful registration, GCM server issues registration id to android device.
3. After receiving registration id, device will send registration id to our server.

4. Our server will store registration *id* in the database for further use.
   a. Whenever push notification is needed, our server sends a message to GCM server along with device registration id (which is stored earlier in the database).
   b. GCM server will deliver that message to respected mobile device using device registration id. (WebGeometrics, 2021)

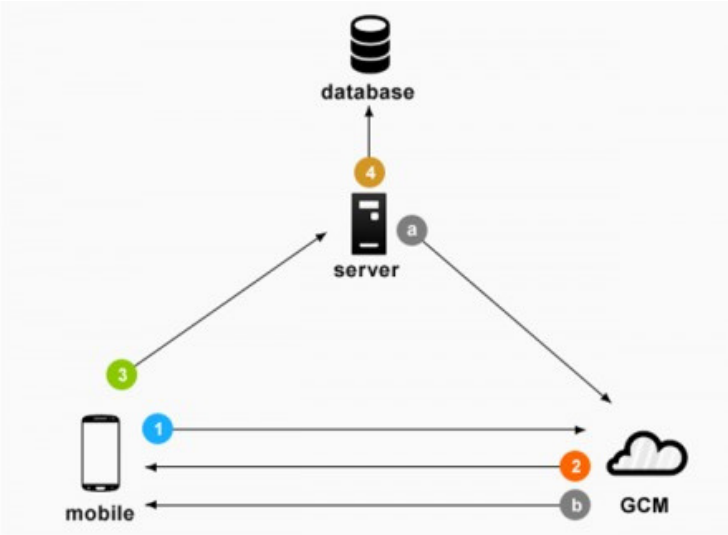


Figure 15. Android push notifications explained

(WebGeometrics, 2021)

3 Related Work

Creating an IoT ecosystem that includes sensors, gateways, cloud services and their respective configurations to reach end users with a web report or mobile application certainly leads to multiple ways to tackle the same problem. This work has taken into account smaller pieces used in other projects and connect them in order to produce a great synergy and reach the goals in a pragmatic and efficient way. Principal works that motivated this project are presented in this section.

3.1 Use case possibilities with Bluetooth low energy in IoT applications [4]

This white paper, apart from explaining the basic notions of Bluetooth Low Energy, shows a big picture of a common flow and propose different use cases for BLE, IoT and the cloud.

In Figure 16. BLE packets broadcast it is shown how the broadcasting mode works. One device broadcast BLE packets periodically. This period, in which two packets are broadcasted, is call advertising time and it plays a fundamental role since it is a trade-off between the periodicity of data to be sent and device' battery consumption. The smaller advertising time, the bigger battery consumption. The observer device is able to receive the advertised packet trigger actions depending on its contain.

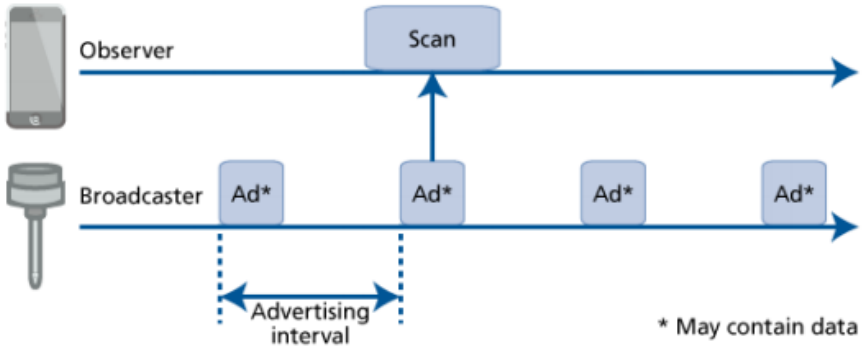


Figure 16. BLE packets broadcast

[4] *https://www.u-blox.com/sites/default/files/products/documents/BluetoothLowEnergy-IoT-Applications_WhitePaper_%28UBX-14054580%29.pdf*

BLE devices can also work in connectable mode as it is shown in Figure 17. BLE connectable mode, where they can expose and set characteristics as it is explained in Bluetooth sub-section.

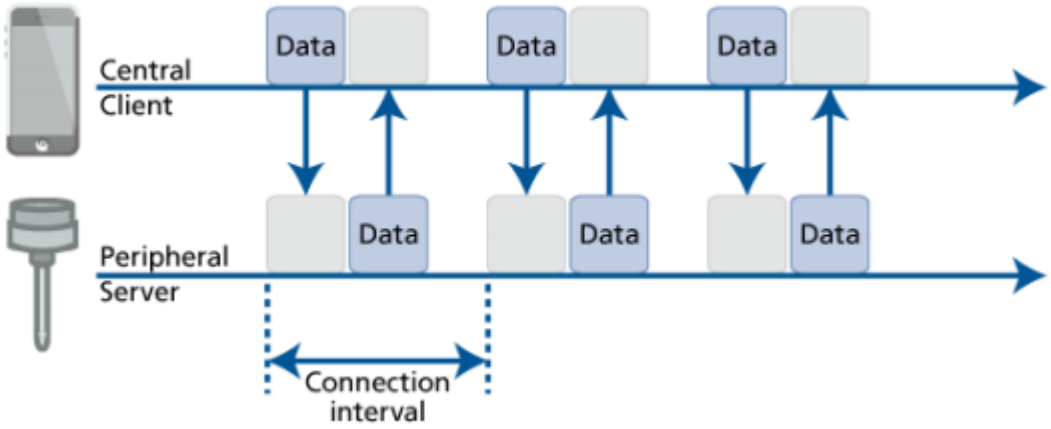


Figure 17. BLE connectable mode

This work also introduces a set of different architectures for BLE IoT projects. In Figure 18. BLE in IoT projects we can distinguish the important role of a gateway in these types of systems. It acts like a “man in the middle” between sensors, which send BLE packets though a GATT service, and the cloud, which process and store data coming from those packets. The author in here propose always to use a RESTful API server, which uses HTTP protocol. However, even though it works, this is not the most effective way of communication, as it will be shown later in *IoT-Based Smart Office System Architecture Using Smartphones and Smart Wears with MQTT and Razberry* section.

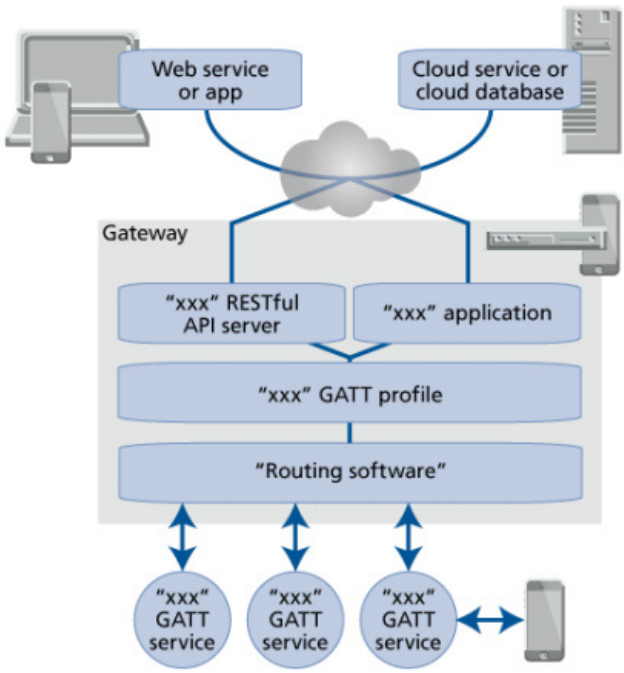


Figure 18. BLE in IoT projects

3.2 Unique Design Challenges of Commercial IoT [5]

It is worth to mention how important gateway are in IoT, especially for commercial purposes. Gateways must know and implement protocols of things, and also of the cloud services. It has to be edge computing capable, flexible, affordable, secure, reliable and with self-healing capabilities. Using an industrial gateway is overcomplicated and expensive. Using a smartphone as a gateway might be considered as well, but it is vulnerable, cannot scale and it is too expensive. Build a gateway with a Raspberry Pi, for example, can also be considered. It is relative cheap and totally customizable, but the downside is that is not scalable at all. For PoC (proof of concept) they are just fine, but it would be hard to manage, deploy and update if the solution faces real world scenarios.

Commercial IoT gateways have best of both worlds, the right mix of edge connectivity, since they implement the protocols needed, and computing capabilities. They are easy to deploy because they can be used out of the box, just with the minimal configuration. They might be more expensive than a Raspberry Pi but not that much as an industrial one.

[5] *https://www.rigado.com/download/whitepaper-unique-design-challenges-of-commercial-iot/?wpdmdl=5606&masterkey=5d1523e20213b*

3.3 Internet of Things for Smart Spaces: A University Campus Case Study[6]

This work is focused on public smart spaces. The authors have provided Tellus Innovation Arena, an open space in the University of Oulu designed for collaborative work, study, and presentation, with a IoT sensor network, using Low Power Wide Area Network (LPWAN), for further data gathering, analysis and visualization. These sensors, strategically positioned, include the measuring of temperature, humidity, CO2, motion, and light.

It is very interesting to see how data analysis of crossing information coming from these sensors may help to take decision and react in smart spaces. This could help to save energy, to have a better meeting room booking service, to improve interactions between people in the space, and provide a better shared place.

Last but not least, authors have made a very clear statement that when it comes to IoT solutions in public spaces challenges are not only technological, like dealing with interoperability, taking IoT from development environments to the industry, massification and scalability. IoT solutions of this type must also deal with ethics, security, and privacy, which is something no to be overlooked.

3.4 User-Friendly and Scalable Platform for the Design of Intelligent IoT Services: A Smart Office Use Case (7)

In this work, authors combine a set of frameworks and tools such as *Dyamand*[8], *Massif*[9] and *Tengu*[10] along with a commercialized wearable, called MiBand[11], to detect presence of people with different roles in the office and trigger specific actions based in theirs profile. This is a great example of interoperability in IoT. (Pieter Bonte)

[6] *https://www.mdpi.com/1424-8220/20/13/3716/htm*

[7] *http://ceur-ws.org/Vol-1690/paper99.pdf*

[8] *https://core.ac.uk/download/pdf/206696619.pdf*

[9] *https://www.researchgate.net/publication/305076700_The_MASSIF_platform_a_modular_and_semantic_platform_for_the_development_of_flexible_IoT_services*

[10] *https://www.tengu.io/dataops-interface/the-tengu-platform*

[11] *https://www.mi.com/global/miband*

3.5 Prototype of smart office system using based security system12

In a study conducted by Pancorowati, researchers found that 83% of employees expected proper lighting, appropriate work areas, and comfortable air temperature. They also expected a convenient archive room, personal office with a neat arrangement of cables. The expected work environment can be seen in the table below (Figure 19. Employee's expectation survey).

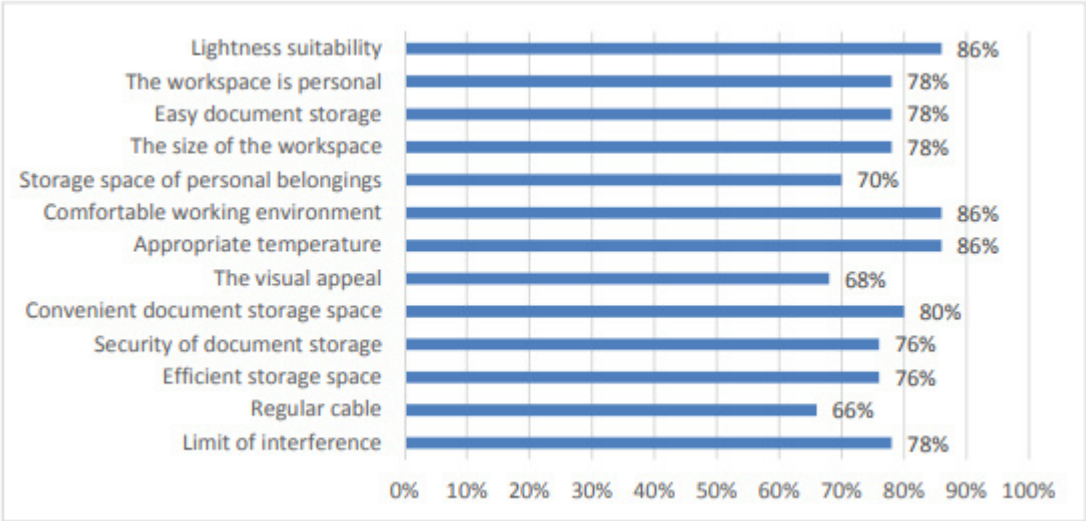


Figure 19. Employee's expectation survey

(T F Prasetyo, 2018)

Authors show how can IoT solution improve office security. They present a system with two different functions. The first one related security mechanism for inside the building, such as dangerous object detector, fire, earthquakes, and theft using specific sensors to detect each action. The second function is related to the security outside the building, using a mobile robot with a camera that patrol the building. This project uses common tools in the IoT world such as Arduino and Raspberry. (T F Prasetyo, 2018)

3.6 IoT-Based Smart Office System Architecture Using Smartphones and Smart Wears with MQTT and Razberry[13]

In their work author use MQTT as a lightweight protocol in a IoT solutions to improve security in smart office. Which is interesting here is that they compared MQTT (Figure 20. Time consumed when using MQTT

[12] *https://iopscience.iop.org/article/10.1088/1742-6596/1013/1/012189*

[13] *https://www.researchgate.net/publication/328078134_IoT-Based_Smart_Office_System_Architecture_Using_Smartphones_and_Smart_Wears_with_MQTT_and_Razberry_ICDECT_2017*

) with AMQP (Figure 21. Time consumed when used AMQP

), concluding that MQTT is a faster approach. (Padmapriya R, 2017)

It is important to note that the Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications or organizations, allowing to connect across organizations, technologies, time and space. (AMQP, 2021)

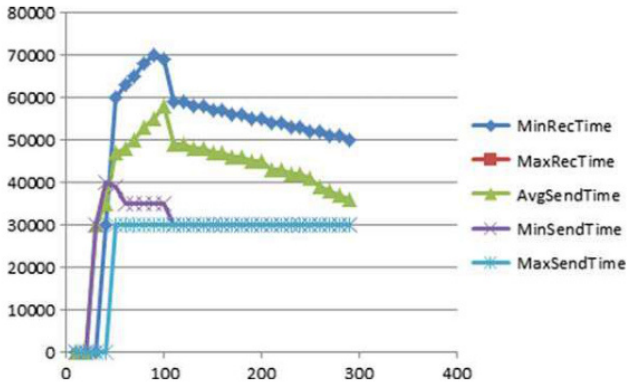


Figure 20. Time consumed when using MQTT

(Padmapriya R, 2017)

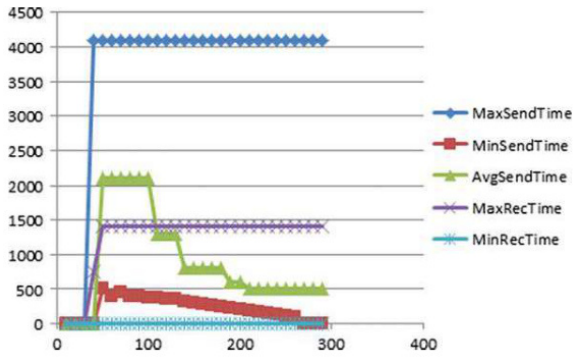


Figure 21. Time consumed when used AMQP

(Padmapriya R, 2017)

3.7 An Innovative Approach for Secured Smart Office and Home System using IoT[14]

In this work, low power and low-cost equipment has been used to create a smart home environment. The team has proven that is possible to provide a good service using just a ESP8266[15] chip instead of using more expensive ones, such as Arduino or Raspberry. All sensors communicate with a control unit layer, which is in charge of gathering the data and

[14] *https://iopscience.iop.org/article/10.1088/1742-6596/1716/1/012056*

[15] *https://www.espressif.com/en/products/socs/esp8266*

send it to the cloud by using HTTP protocol. Then a mobile application communicates with the cloud to get the data gathered also using HTTP protocol. System architecture is shown in Figure 22. An Innovative Approach for Secured Smart Office and Home System using IoT. System architecture.

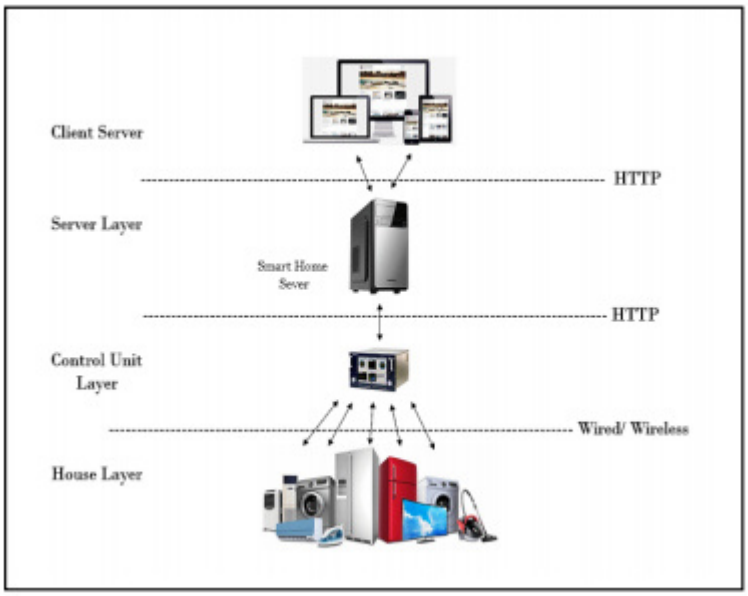


Figure 22. An Innovative Approach for Secured Smart Office and Home System using IoT. System architecture.

4 Architecture

In order to achieve the goal planned in this work, and being it an IoT solution, a set of technologies, systems, cloud services, communication protocols and devices need to interact with each other. It is proposed the architecture shown in *Figure 23. Architecture* to describe the interaction between modules and each of them will be explained next, together with the flow of data.

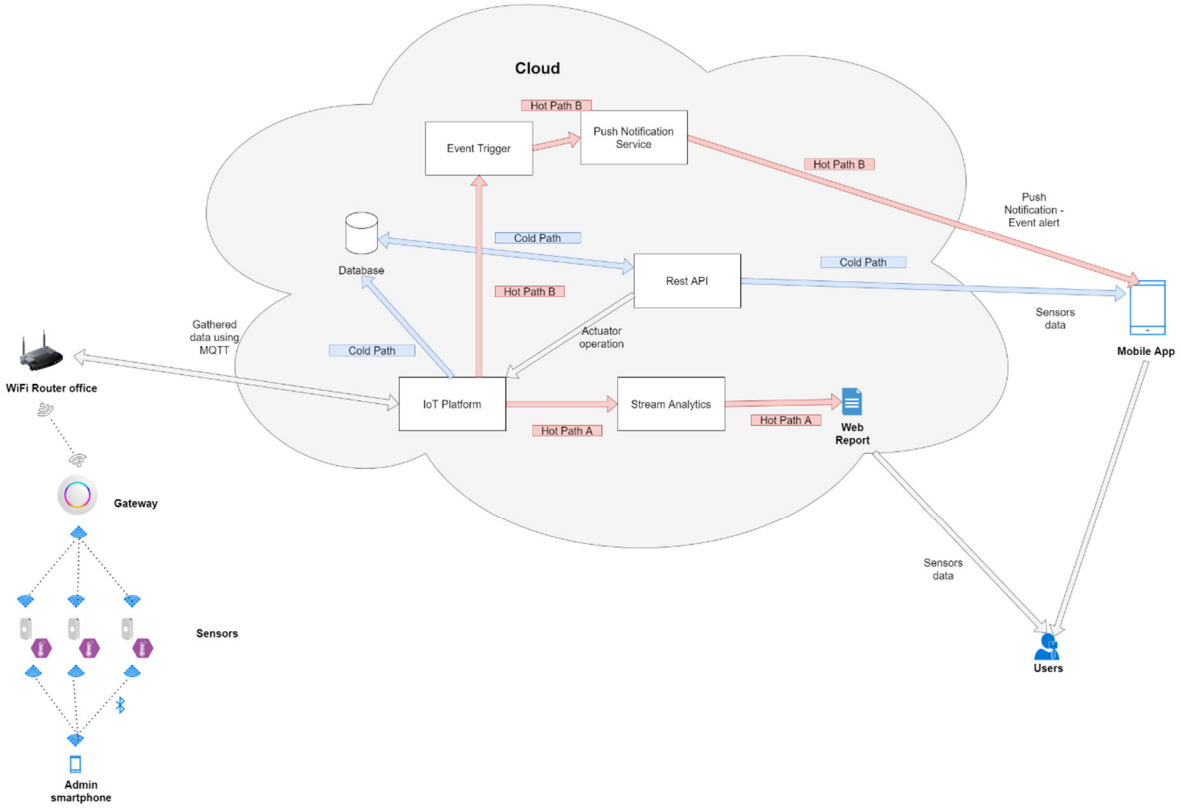


Figure 23. Architecture

4.1 Sensors

As it was mentioned in *Smart device / sensor layer* section, in this project, each sensor will get a big amount of information from the smart office and will send it to the office *gateway*.

There is no restriction of what type of sensor can be used as long as it implements BLE protocol. That means that sensors will be able to broadcast packets, in this case with relevant data. Sensors can also broadcast packets of its health or current status.

When using Bluetooth, sensors can also enter the connectable mode, allowing other BLE devices to change its configuration. Advertising time, type of packets, and their content to be broadcasted need to be configured for each device.

4.2 Admin smartphone

Bluetooth capabilities of current market's smartphones will be harnessed to connect it with each sensor. By this way, we can get and send GATT characteristics to each BLE device. This allows administrator users to change sensors configuration, mentioned before.

4.3 Gateway

The main function of the gateway is getting data of all sensors in the smart office and send them to the IoT platform. It is very likely that in the office will be diverse Bluetooth devices, such as employees' smartphones, wearables, notebooks, headphones, beacons, among others BLE devices. Gateway needs to be able to know which packets are from the proper devices and filter them, to send to the cloud only the necessary information and no more than that. Otherwise, it can have an impact not only in performance and readability of data but also on costs.

To send filtered Bluetooth packets with sensors data to the IoT platform in the cloud, office's gateway must implement MQTT protocol and know beforehand which topics are needed to be subscribed and publish on them.

Last but not least, to be able to communicate with any service in the cloud gateway need to be connected with the current office WiFi Router so that it can have access to internet.

4.4 IoT Platform

This module needs to implement MQTT protocol with the aim of receiving sensors data coming from the gateway. This connection must be secure. The IoT platform can retrieve information from multiple offices using the same MQTT channel. It should be possible to subscribe to that topic an see real time events. It will communicate with the database to store data for further purposes and with the streaming analytics for real events analysis. These are the cold and warm paths, respectively.

4.5 Database

Database plays the role of persisting sensors data for further analysis. In this scenario, it will be accessible by the Rest API, but other new services can be built using this database.

4.6 Rest Web API

This module consists in a web API that reads data from database and implements HTTP/S protocol to expose secure endpoints for client's application in order to execute

GET, POST and PUT operations according to the system needs. This is also a key for users to configure and set thresholds in their own profile. It can also be used by other client applications.

4.7 Event Trigger

The event trigger is a process that continuously runs analysing data from logs, databases or other data sources, searching for rules, thresholds or specific criteria to be meet. Once it realizes that this was achieve, it communicates to other modules of what have happened. For example, the event trigger can analyse temperature data in the database, to be bigger than 25ºC. When that occurs, it generates and alert and send a notification to the user (for instance, sends an email or a push notification)

4.8 Push Notification Server

The push notification server receives the alert from the event trigger and is in charge of sending that alert in form of a real time push notification to all users with the mobile application installed in their smartphone who have been subscribed, even if they are not using the application at that moment.

4.9 Mobile Application

The mobile application uses Rest API endpoints to get sensors data from database and show current conditions of the office. It can be used from anywhere, at any time as long as the users have internet connection and the permissions to connect. It will also install a service in user's device to receive push notification in real-time.

4.10 Streaming Analytics

When using hot path, streaming analytics service will analyse IoT Platform events in real time. Along with the event trigger and the push notification server, it plays a fundamental role in bring a real time alert to the users of what is currently happening in the smart office.

4.11 Web Report

Using cold path, sensors data will take some seconds and maybe a minute to save data in the database. This is not useful when real time alerts are needed but is a cheaper way for users to able to get data from past events. For example, get the temperature and lightning in January and compare it with February. Multiple web reports can be created depending on the needs of the users. They will get data from database and show them to the user in a browser. Different sorting, grouping, selecting and other data operation will be performed here to show users the information they want to see as clearer as possible.

4.12 Sensors' data flow

In the proposed architecture there are three paths to reach to send sensors data from the IoT platform to the end user. These ones are described below.

1. **Cold path**

IoT platform persists sensors data in the database. *Rest API* connects with the database to get this information when users make use of the *Mobile Application.*

2. **Hot Path A**

Stream Analytics listens to *IoT platform* events on real-time and create the dataset so the *Web Report* can show that information instantly.

3. **Hot Path B**

Event Trigger listens to IoT platform events on real-time and send a message to the Push Notification Server. This last one, analyse the alert and send a push notification to all users where the rule applies. At the end, users receive a push notification in their mobile phone instantly.

5 Implementation

For the implementation, it has been chosen and employed a technology for each module described in the previous section which allow modules to accomplish their function and communicate with the rest. Technologies for each module and its interaction can be seen in Figure 24. Project implementation diagram

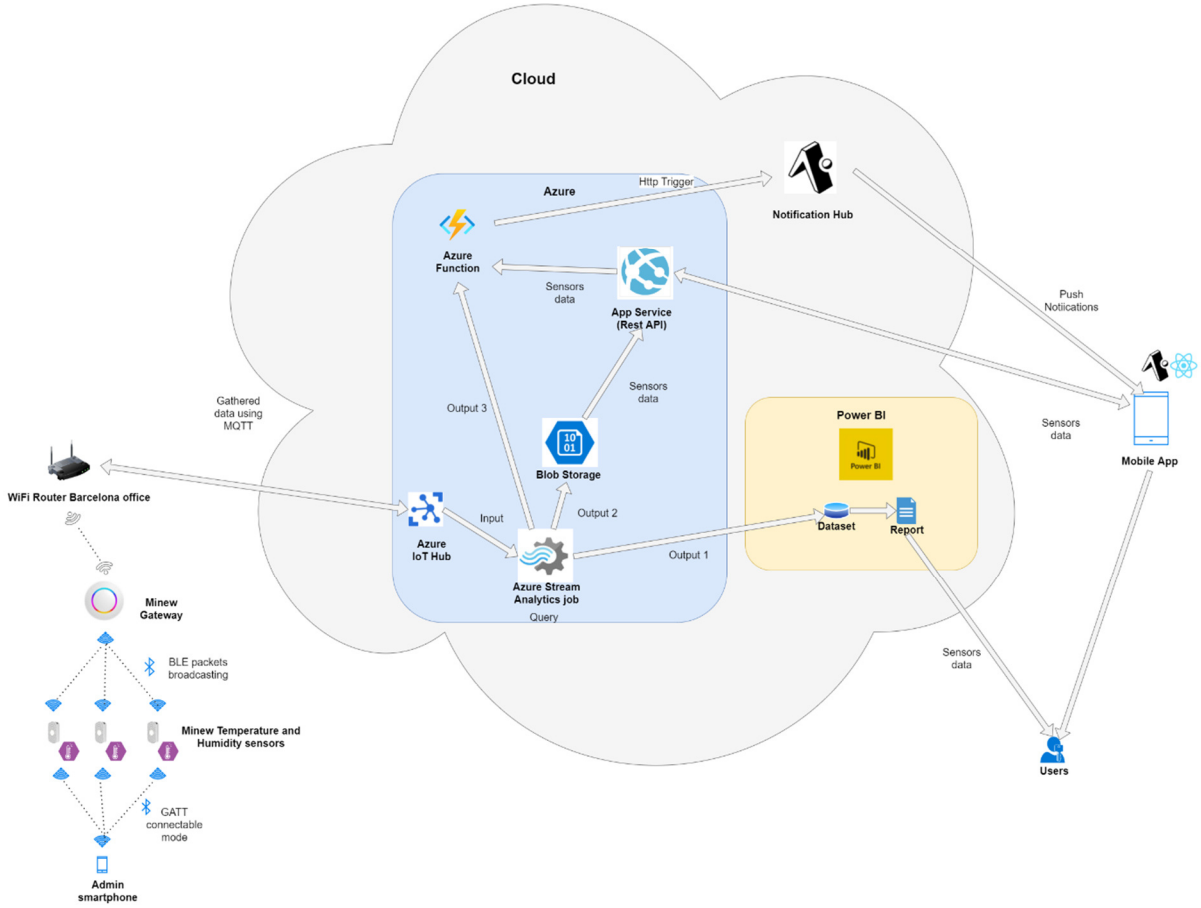


Figure 24. Project implementation diagram

5.1 Sensors & Admin Smartphone

To measure temperature and humidity in their office, iSolutions has been purchased three Minew S1 sensors (Figure 25. Minew S1 Temperature and Humidity sensor). These are BLE beacons with temperature and humidity sensor and accelerometer. They are compatible with both iBeacon and Eddystone (UID, URL, TLM) protocols and they can convert the environmental temperature and humidity into correspondent digital signal and

store up to 200 history records. Cost of each device in March 2021 is $20 (USD). Some extra features they offer are [16]:

- IP65 waterproof level.
- Over 100 meters broadcasting range.
- Over 3 years battery lifetime.
- Ultra-low power consumption chipset nRF52832.
- High-accuracy digital temperature and humidity sensor inside.
- Very fast start-up and measurement time.
- Real-time data converting, storing and uploading.
- Useful security mechanism. To access to its configuration, users need a password and, furthermore, connectable mode can be disabled at a touch of a button, meaning no one without a physical access to it can read and changes its settings.
- Supporting the customization service such as logo, firmware and other.

Figure 25. Minew S1 Temperature and Humidity sensor

User manual of this device has been carefully studied and analysed and can be consulted in a personal shared folder[17], along with all its specifications.

Administrators of the office uses their smartphone BLE capabilities to configure each sensor device. In this work, an Android device was used. In order to achieve that, an application developed by the manufacturers has been used, called *BeaconSET+*[18].

[16] *https://en.minewtech.com/sensor/S1.html*

[17] *https://drive.google.com/drive/folders/1UUFl3Ege1vfriHCry0Y5ClVgeeEW3ZQ-?usp=sharing*

[18] *https://play.google.com/store/apps/details?id=com.minnw.beaconset&hl=en&gl=US*

Once the application is open, it will show a list of devices along with some data that is being broadcasted. In figure Figure 26. BeaconSet+. List of devices, a list of the three sensors located in strategic areas of iSolutions Barcelona office is displayed.

Figure 26. BeaconSet+. List of devices

In the office's maps shown in Figure 27. Distribution of devices at the office I and Figure 28.Distribution of devices at the office II. Room 26, location of the S1 sensors and the G1 gateway are indicated. Distance between D28 (main office) door and D26 door is about 8 meters.

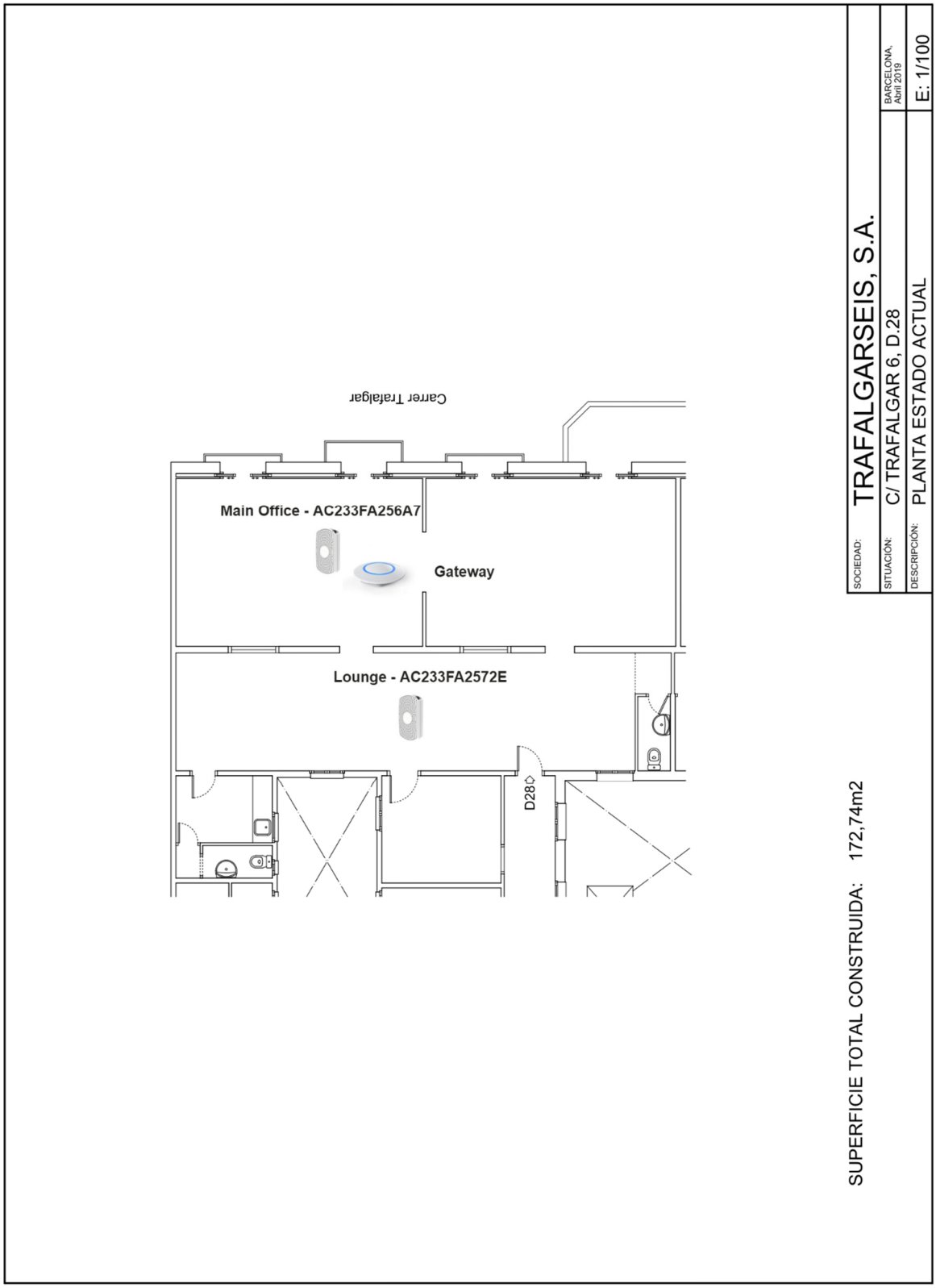


Figure 27. Distribution of devices at the office I

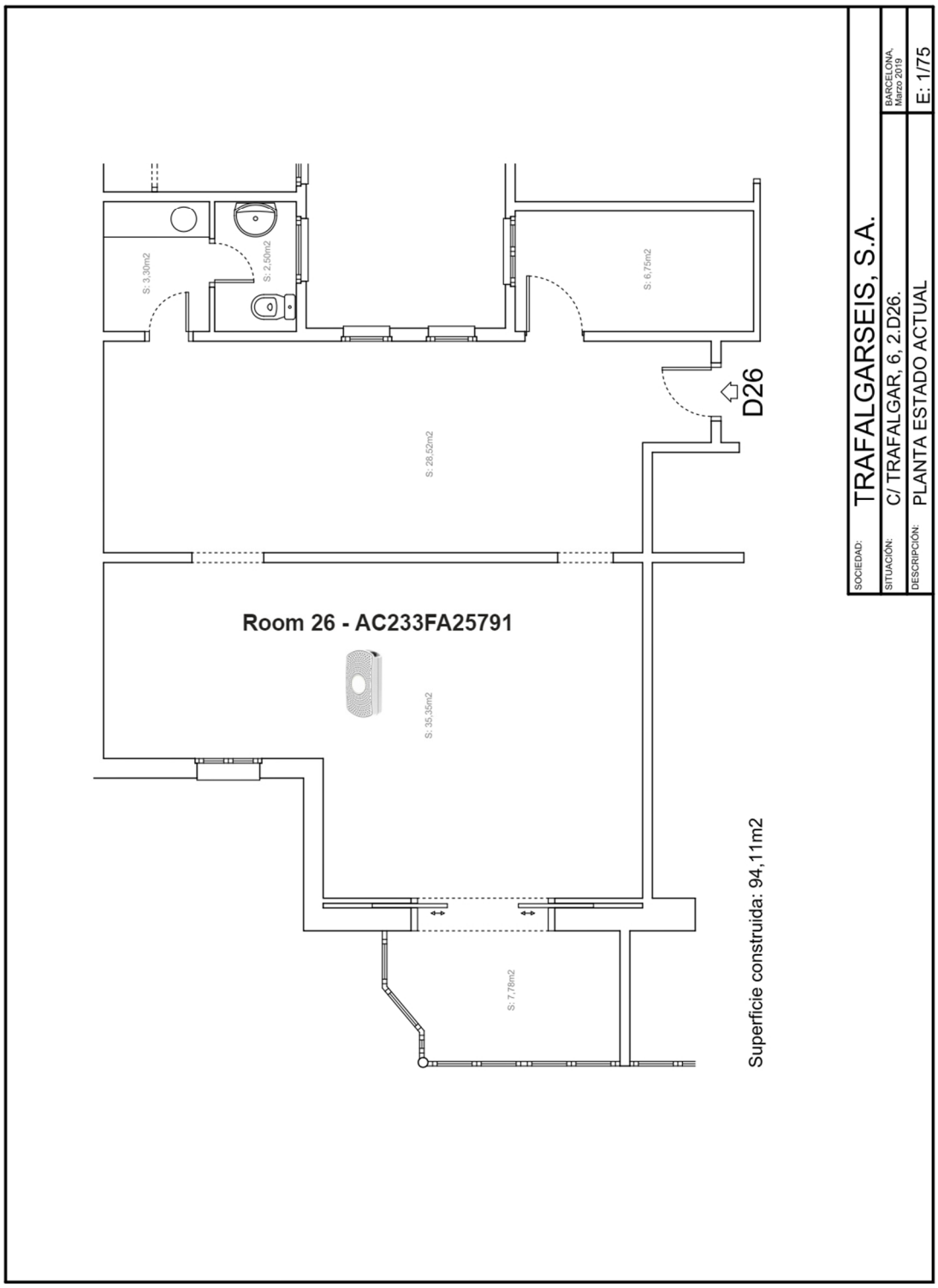


Figure 28.Distribution of devices at the office II. Room 26

Sensor devices, for performance and low energy consumption reasons, are configured to not be in connectable mode. However, it is possible to activate the connectable mode by pressing the only button they have. After that, a green circle will be shown in the list to notify the users which sensors are in connectable mode (Figure 29. BeaconSet+. A device with connectable mode on). By tapping the screen and providing the credentials (Figure 30. BeaconSet+. Password required to connect to a sensor), admin users can access and configure each device.

Figure 29. BeaconSet+. A device with connectable mode on

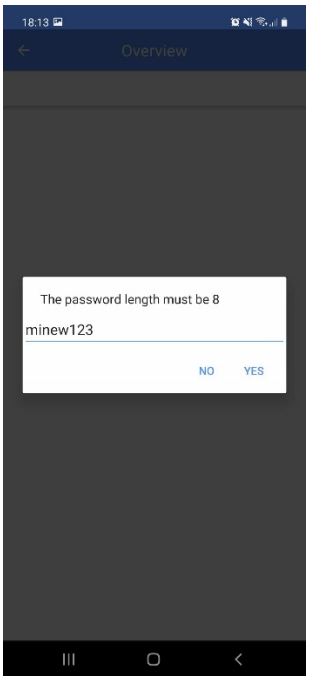


Figure 30. BeaconSet+. Password required to connect to a sensor

Once user connects to a S1 sensor using BeaconSet+, it is possible to:

- Turn on/off connectable mode.
- Update firmware.
- Reset device to default settings.
- Turn it off.
- Remove and change password.
- Read manufacturer and feature info.
- Synchronize humidity and temperature (HT) data to be store in local phone and see It in a chart.
- Configure slots with type of data that needs to be broadcasted.

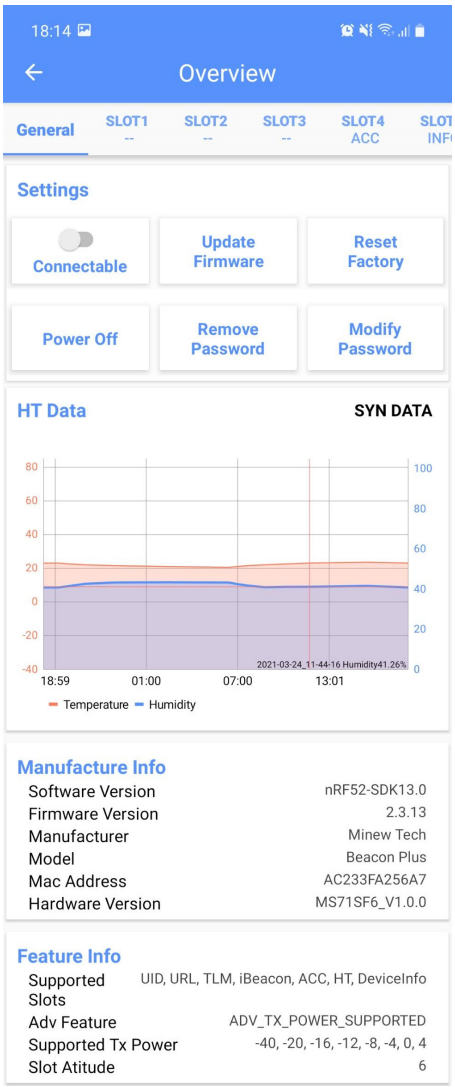


Figure 31. BeaconSet+. Connected to a S1 sensor

Apart from sending humidity and temperature data (HT), a S1 can also broadcast: Eddystone packets (UID and URL), TLM, iBeacon, ACC (motion data) and DeviceInfo packets. A device can be configured to send all of the packets, or any combination of them using the 6 available spots.

In order to optimize battery life and network usage, only HT (Figure 33. BeaconSET+. HT configuration) and DeviceInfo (Figure 32. BeaconSET+. DeviceInfo configuration.) packets, this last one recommended by the manufacturer to be better recognized by the Gateway, are used in this project.

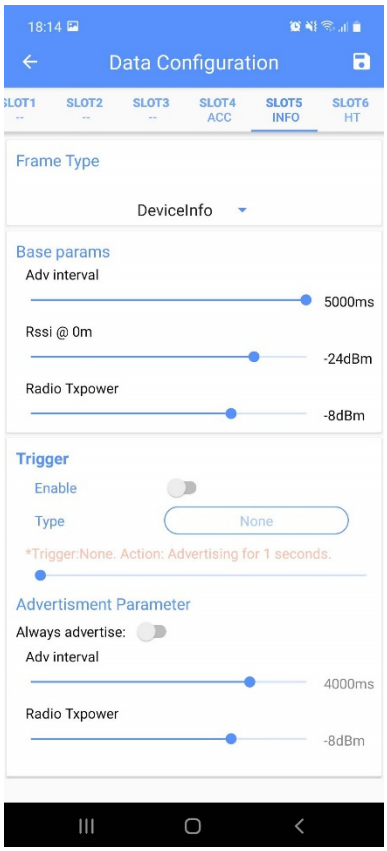


Figure 32. BeaconSET+. DeviceInfo configuration.

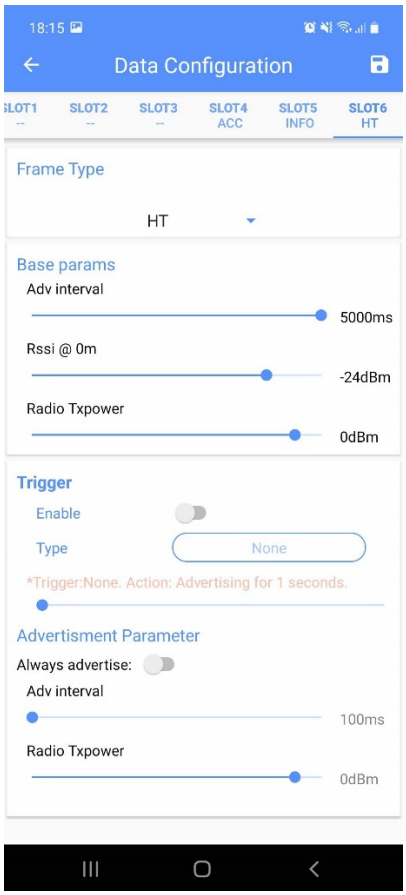


Figure 33. BeaconSET+. HT configuration

For each frame type, it is possible to configure different sort of parameters. For instance, for broadcasting an URL of Eddystone protocol, we can set the URL to be broadcasted. However, there are configuration parameters shared between all of them, such as the advertising time, which was set on 5 seconds for optimization reasons, knowing that HT values will not drastically change in a short amount of time.

Other values are Received Signal Strength Indicator (RSSI), a measure of the power level at the receiver, and transmission power, which also play an important role in the communication with the gateway and batter consumption.

S1 sensors allow users to set triggers, to send data only when some conditions are meet. For example, only send HT data if the temperature is below 15°C, or when motion values are high (which implies that the sensor fell from the wall, or someone is manipulating it).

5.2 Gateway

A device from the same brand is used as a gateway, called Minew G1 (Figure 34. Minew G1 Gateway). It works as a BLE to Wi-Fi gateway. It also can be configured to be a Wi-Fi access point. G1 is a Bluetooth 5.0 low energy to Wi-Fi connectivity gateway without the uses of smartphones or apps. The G1 gateway collects the data from iBeacon, Eddystone,

BLE sensor and other BLE devices, and then sends to the local server or remote cloud server by HTTPs/MQTT/mbed (ARM) protocol with SSL and TLS certification over Wi-Fi , ethernet or cellular. Some of the features it offers out of the box are the following:

- Monitor BLE beacon
- BLE beacon configuration control
- Management BLE beacon
- Upload BLE beacon data to IoT cloud platform
- Analytical identification BLE beacon data.

Other feature to highlight is that it can operate in a range of 300 meters in an open space. For further information and technical details, the user manual, data interface and configuration guide are also shared in the personal shared folder [19].

Figure 34. Minew G1 Gateway

Figure 35. Minew G1 in action, taken from user's manual, shows how G1 gateway interacts with other BLE device sensors of the same brand, though it can receive BLE packets from any other brand, and transmit them to the cloud via WiFi using different protocols. Exactly the set-up proposed in iSolutions Barcelona office, which has been presented in the Architecture section.

[19] *https://drive.google.com/drive/folders/1UUFl3Ege1vfriHCry0Y5ClVgeeEW3ZQ-?usp=sharing*

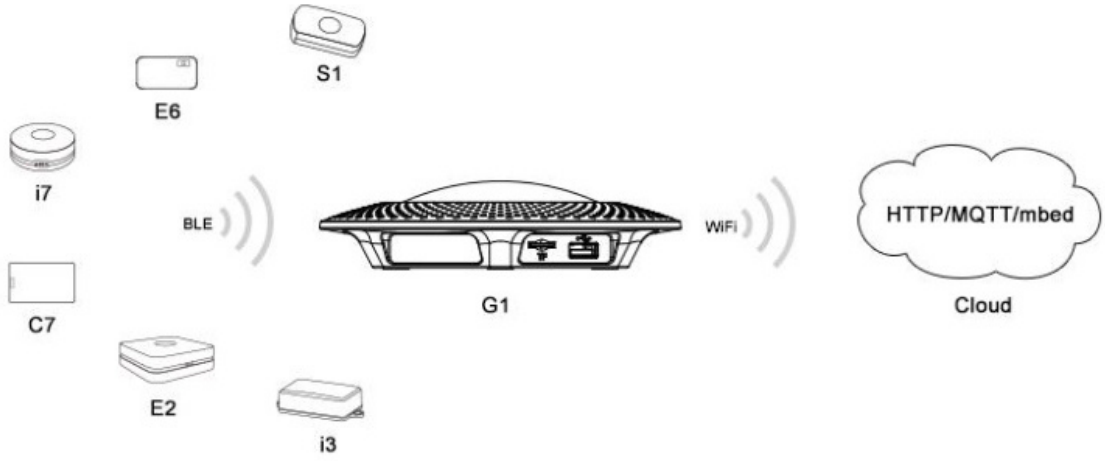


Figure 35. Minew G1 in action

In order to configure the gateway, it is needed to establish a connection. It can be done using Ethernet or Wi-Fi. In this work, It the connection from a computer to the gateway was made through Wi-Fi. By default, G1 has the access point open. Once one is in connected in the network it can be accessed by using the local IP *192.168.99.1* in any browser. The configuration gateway webpage will be displayed, and it will ask for credentials (Figure 36. Minew Gateway G1. Login).

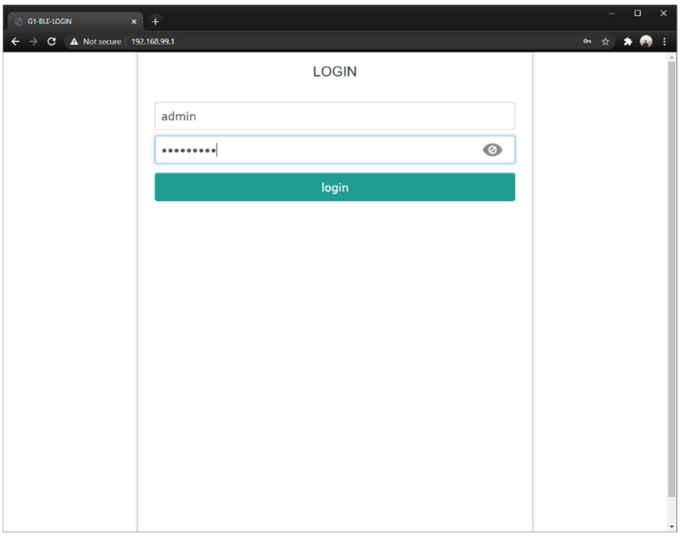


Figure 36. Minew Gateway G1. Login

When user is logged in, there are four tabs to configure the device. These ones are:

1. **Status** (Figure 37. Minew Gateway G1. Status view.)

It shows general information about the status of the device, such as firmware version, MAC and current IP. We can also configure here the access point name (Figure 38. Minew Gateway G1. Access point connection), IP and password (if we want it to work as a

repeater) and to set a new admin password for the device (it is recommended not to use manufacturer's default).

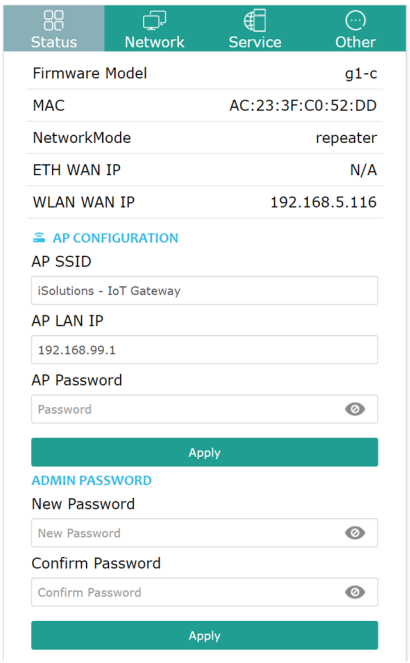


Figure 37. Minew Gateway G1. Status view.

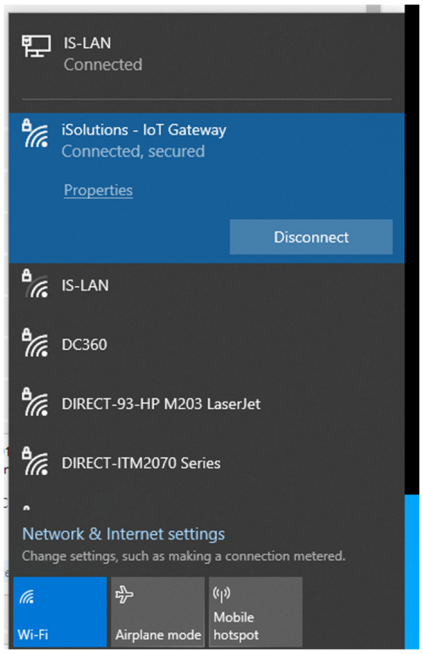


Figure 38. Minew Gateway G1. Access point connection

2. **Network** (Figure 39. Minew Gateway G1. Network view.)

This tab is an important step to configure the output to internet. It may be done by ethernet (wired connection, "router" mode) or connecting to a WiFi router (wireless, "repeater" mode). In this work, the repeater mode has been used, therefore, SSID and

password to the WiFi router of Barcelona Office had to be provided. DHCP mode has been used to let the router to assign the WiFi an IP dynamically.

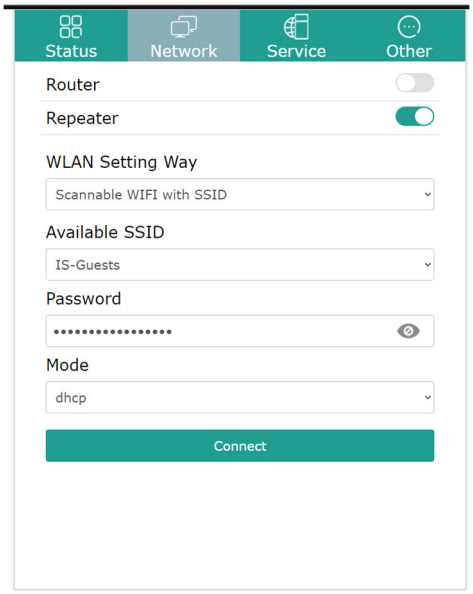


Figure 39. Minew Gateway G1. Network view.

3. **Service** (Figure 41. Minew Gateway G1. Service configuration view.)

Service tab allow users to configure how the communication with the cloud is going to be, so it expects the parameters depending on the type of protocol selected (HTTP or MQTT). As it was mentioned before, for it benefits and advantages over HTTP, MQTT is used.

We can set how often do we want telemetry data to be sent, the URL, which was taken from Azure IoT platform and also, security parameters.

To ensure the connection from the Gateway to the cloud is secure, the cloud service needs to know who the gateway allowed to send data is. For that purpose, it was used a self-signed certificate, which can be uploaded to the Gateway using a USB memory card.

For easing development and avoid costs, a development self-certificate has been created using *openssl*[20].

openssl req -newkey rsa:2048 -nodes -keyout mydevkitkey.pem -x509 -days 365 -out mydevkitcertificate.pem

[20] *https://www.digicert.com/kb/ssl-support/openssl-quick-reference-guide.htm*

For further purpose, the fingerprint was captured by using:

```
openssl x509 -in mydevkitcertificate.pem -fingerprint -noout -sha256
```

This fingerprint will be used in the IoT platform to validate the certificate.

Last but not least, a certificate provided by Microsoft was used[21]. It allows the connection with Azure IoT. We just needed to copy from that file the proper certificate and create a new PEM file using that content.

Figure 40. Microsoft certificate used [22]

With the transmission being secure, the next step is to provide the MQTT parameters expected by the Azure IoT platform, such as username, password, publish and actions topic (to send but also to receive data from the cloud using those specific topics). It is worth to

[21] *https://github.com/Azure/azure-iot-sdk-c/blob/master/certs/certs.c*

[22] *https://docs.microsoft.com/en-us/answers/questions/63722/device-to-cloud-using-mqtt-self-signed-certificate.html*

mention that information about MQTT protocol for Azure is really hard to find, and also, that these topics name, along with the URLs are fixed by the platform and could not be changed.

In addition, other properties that are very important to set are related to the packages to be sent. First, we can set the packages to be sent by RSSI (power), BLE name or MAC. For example, in this work it is used a regular expression to only set data provided by the S1 devices. This avoid data issues on the IoT platforms, because, as it is known, there are a lot of BLE devices in the office and the gateway can also receive these packets. At the very bottom we can also select what type of packet we want to send to the cloud. In this case, we do not need to send iBeacon packets since they are not use, for instance. It is a good practice to filter the packets as much as possible and only send the information that is needed at the time that is needed.

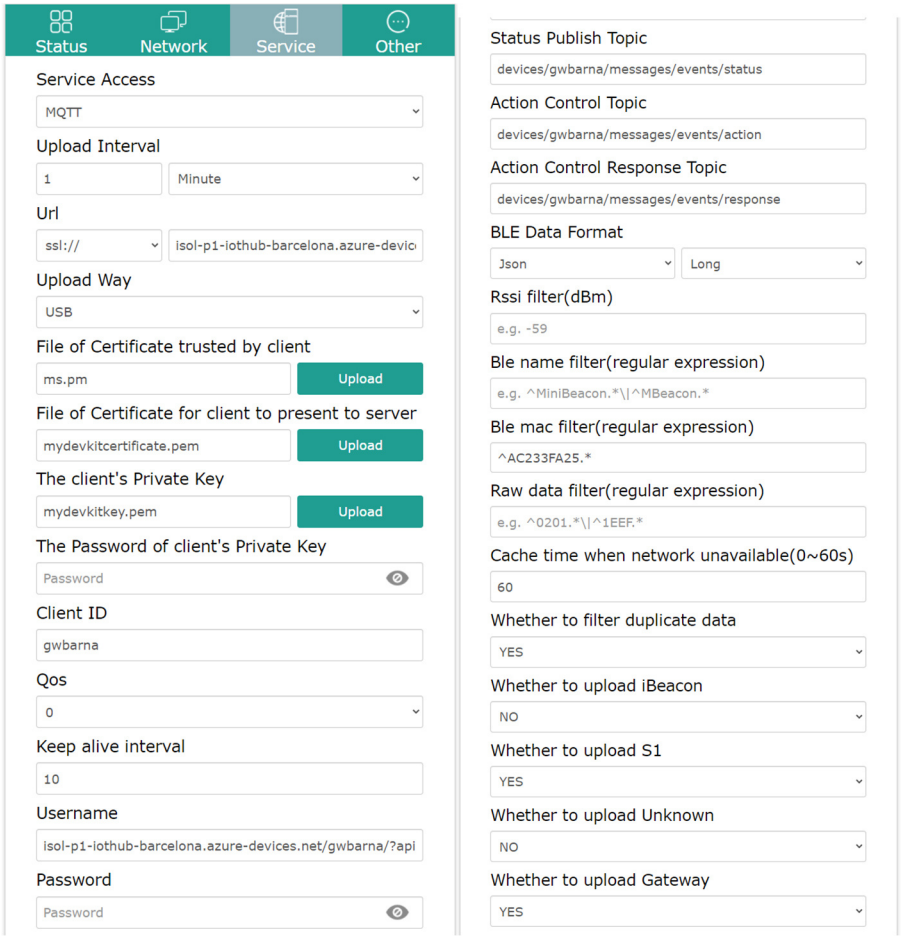


Figure 41. Minew Gateway G1. Service configuration view.

4. **Other** (Figure 42. Minew Gateway G1. Other configuration)

The last configuration tab is meant to enable/disable device lights, set time zone and NTP servers (to send packets with the proper date and time on it), reboot, upgrade firmware, enable/disable BLE and set its interval time, reset from factory and last the option to turn or/off the access point. For security reasons, once the gateway was configured, the access point has been turned off, so nobody can connect to it unless someone hold the reset button for two seconds using a very thin tool.

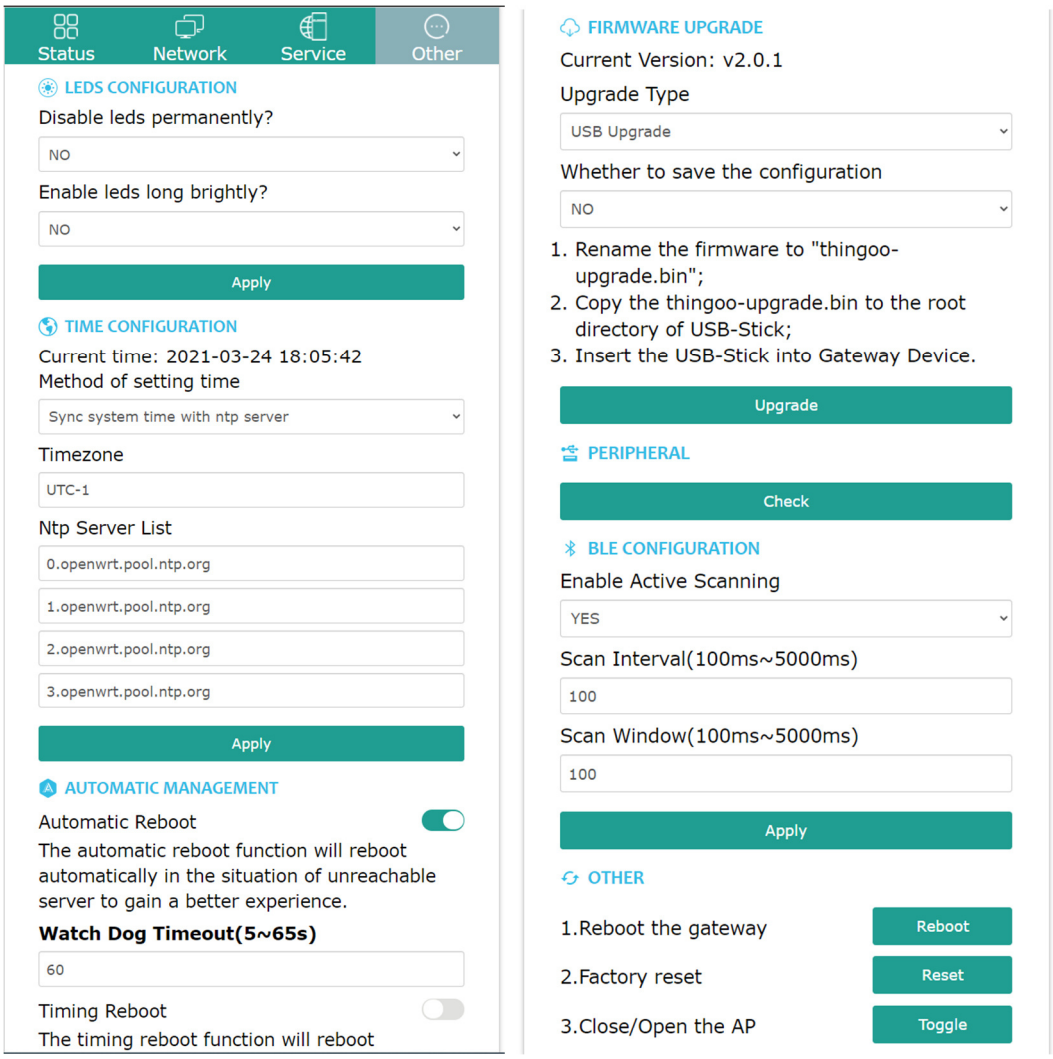


Figure 42. Minew Gateway G1. Other configuration

5.3 IoT Platform

In terms of IoT cloud platform, Azure IoT Hub was chosen over Azure IoT Central not only due to same capabilities at a minor cost but also because it provides a more configurable platform, as it is destined to more technical professionals.

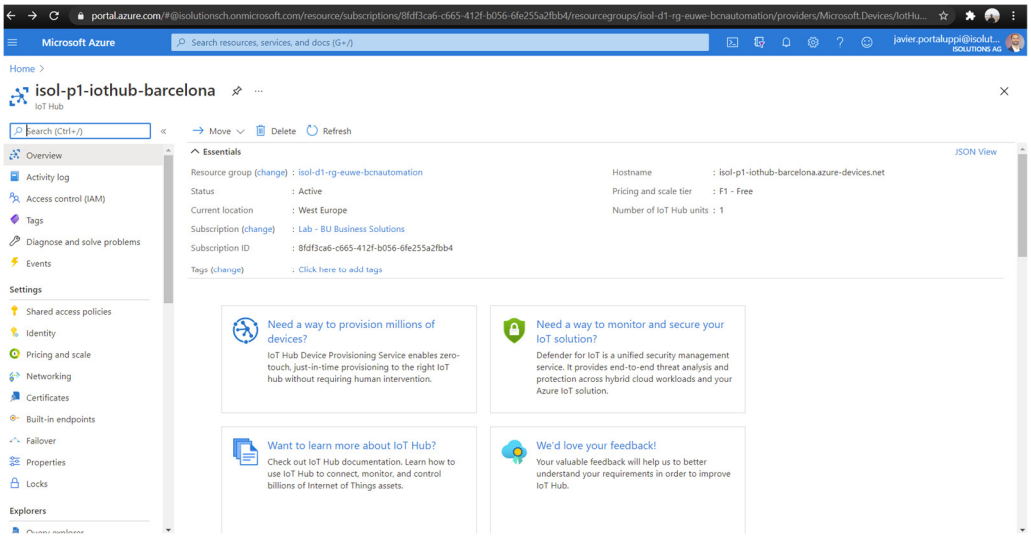


Figure 43. Azure IoT Hub Dashboard

Azure IoT Hub listen MQTT request on the following topics, as explained in “Using the MQTT protocol directly (as a module)” section of official documentation[23]:

- devices/{azureDeviceName}/messages/events/status
- devices/{azureDeviceName}/messages/events/action
- devices/{azureDeviceName}/messages/events/response

Only one device was added in IoT hub, which will virtually represent the gateway in this case, and it is named “gwbarna”, allowing the communication between the cloud platform and the gateway ().

[23] https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support

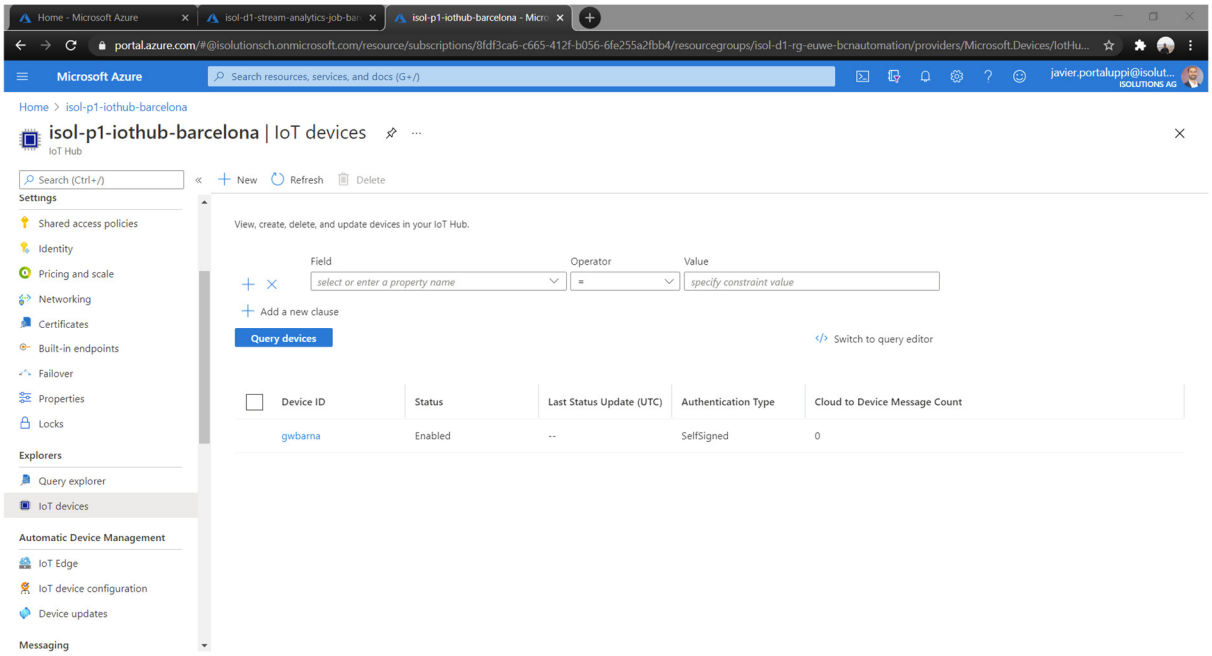


Figure 44. Azure IoT Hub. IoT Devices

For security reasons, this device has been created using the fingerprints obtained in previous section Gateway.

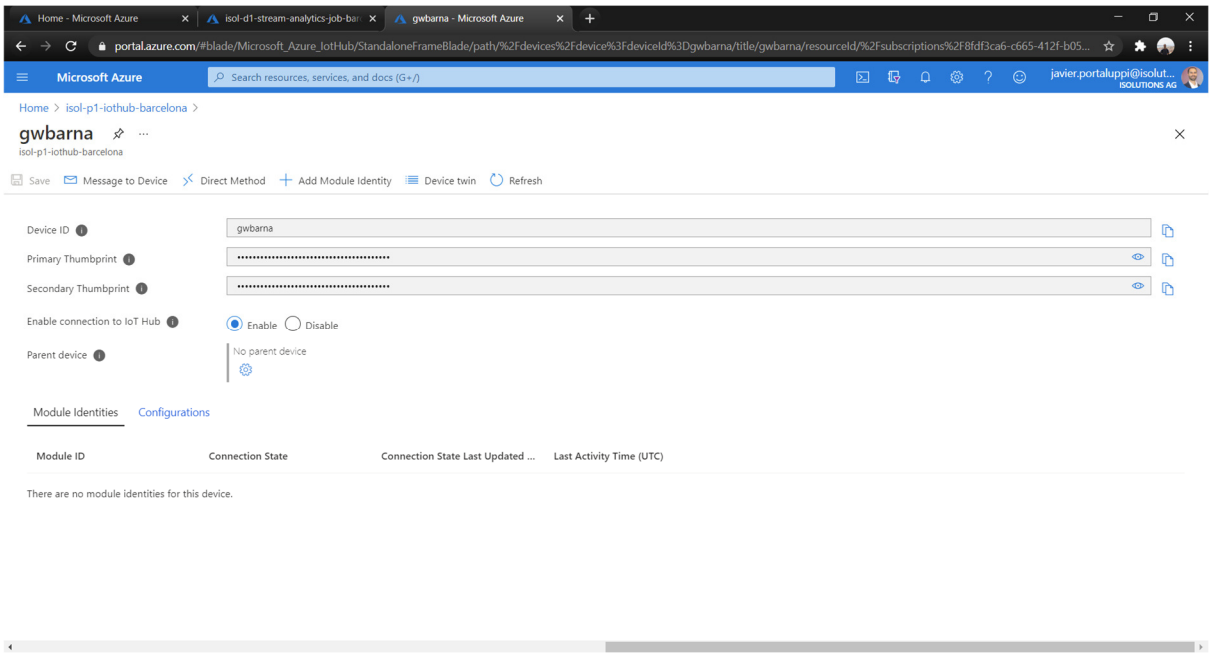


Figure 45. Azure IoT Hub. gwbarna device detail

5.4 Database

A document database (no SQL) was chosen over a relational one (SQL) due to the type of solution it is being developed. It gives more flexibility when it comes to persisting different fields from different sensors. For example, In the same document, it is possible to store temperature and humidity data but also luminosity. As it was mentioned before,

interoperability is one of the keys in IoT solutions. Hence, a *Cosmos DB* database[24], called *BcnOfficeIoTDatabase,* was configured and used for this with two tables:

5.4.1 S1 Sensor table

This table is loaded by the event hub that listens telemetry data on IoT hub and save it here. Therefore, each contains data referring to the current temperature and humidity, the sensor MAC, and datetime it was measured.

For development easing, in this work this table is loaded using the Azure Streaming Analytics jobs, meaning that this table will always have real time data but at a pricey cost.

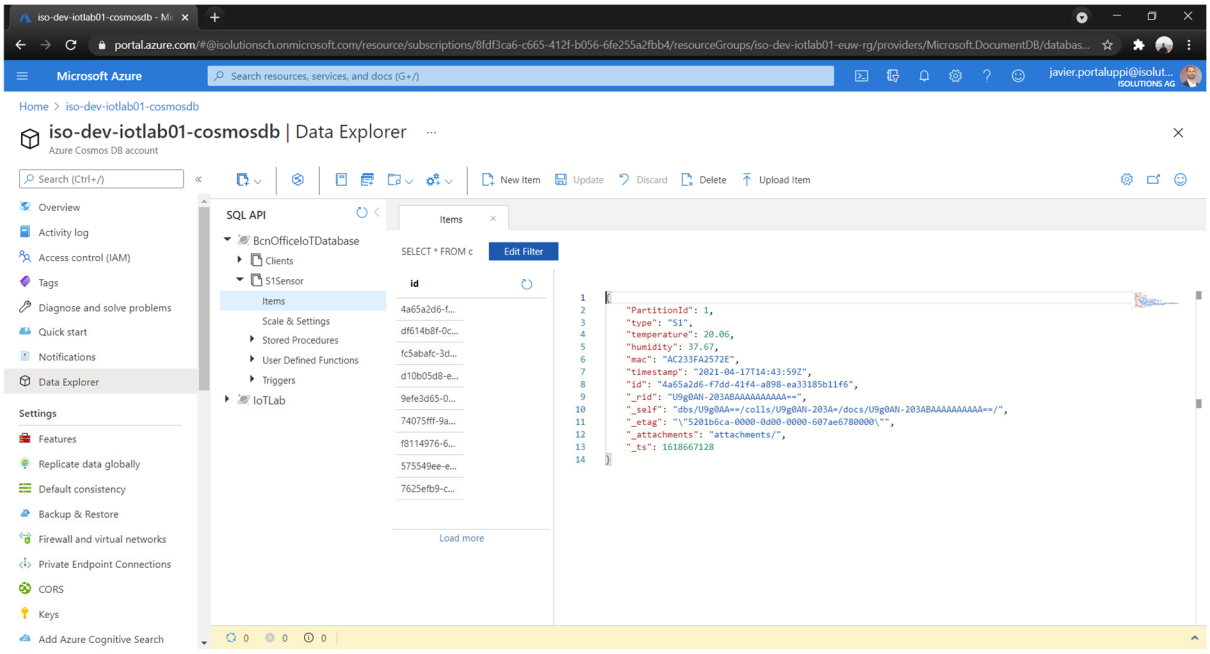


Figure 46. SensorS1 table in Cosmos DB database

5.4.2 Clients table

This table is fundamental for making this solution scalable. For only one client that would not make any sense, but if we want to scale up or deploy this solution to multiple office and multiple tenants we would need a table with a user identification, which are the devices/sensors that are registered for that client, which are its preferences (for instances, thresholds and other alert preferences/parameters) and if it active or not (for example, to deactivate a user if he did not pay the service)

[24] *https://azure.microsoft.com/en-us/free/cosmos-db/*

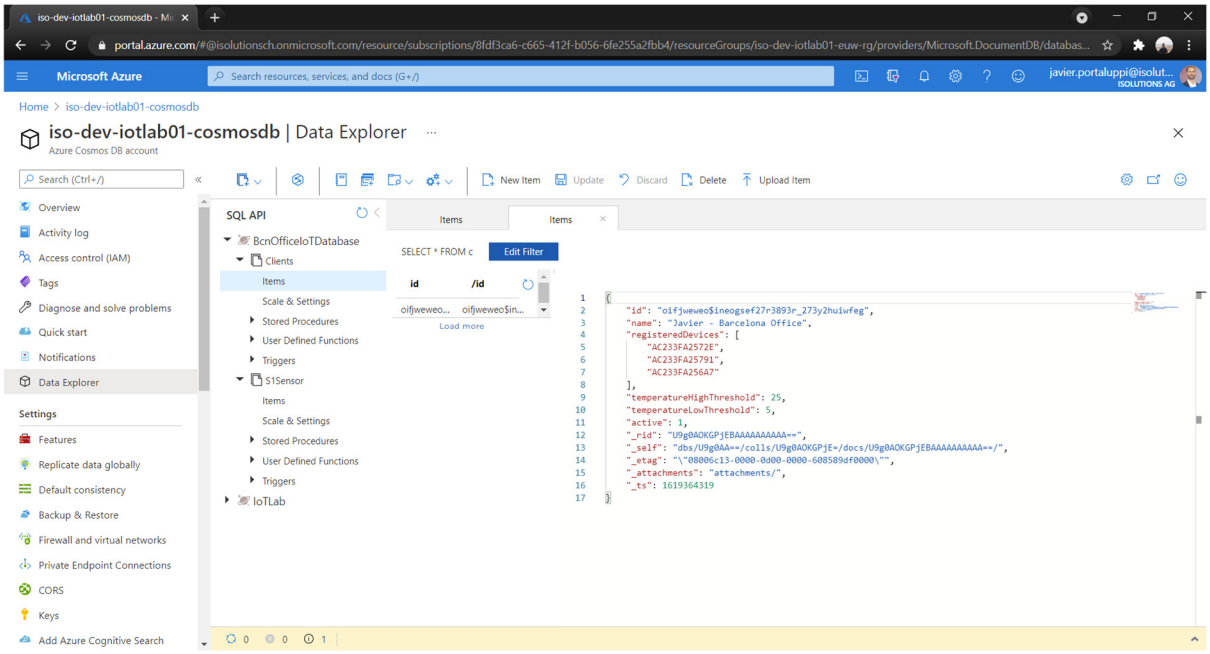


Figure 47. Clients table in Cosmos DB database

5.5 Rest API

This module consists in a .Net Core web API (programmed in C#) that reads data from database and implements HTTP/S protocol to expose secure endpoints for front end applications to visualize sensors data (GET operations). In addition, this system can also expose endpoints to write (POST operations), in order to either manually trigger some events or turn on an actuator in the office, meaning that the Rest API can also be connected with the IoT Platform, and PUT actions as well, to update user parameters and preferences.

This API consists in 3 layers:

5.5.1 BcnIoTWebAPI

This is the main project, and its main functions are exposing the API controllers and call the services layer. There is one controller for client's resource (ClientsController.cs) and other one for the S1 sensor data (TemperatureAndHumiditySensorsController.cs), In addition, it implements swagger[25], a UI to see, document and try API controllers.

[25] *https://swagger.io/*

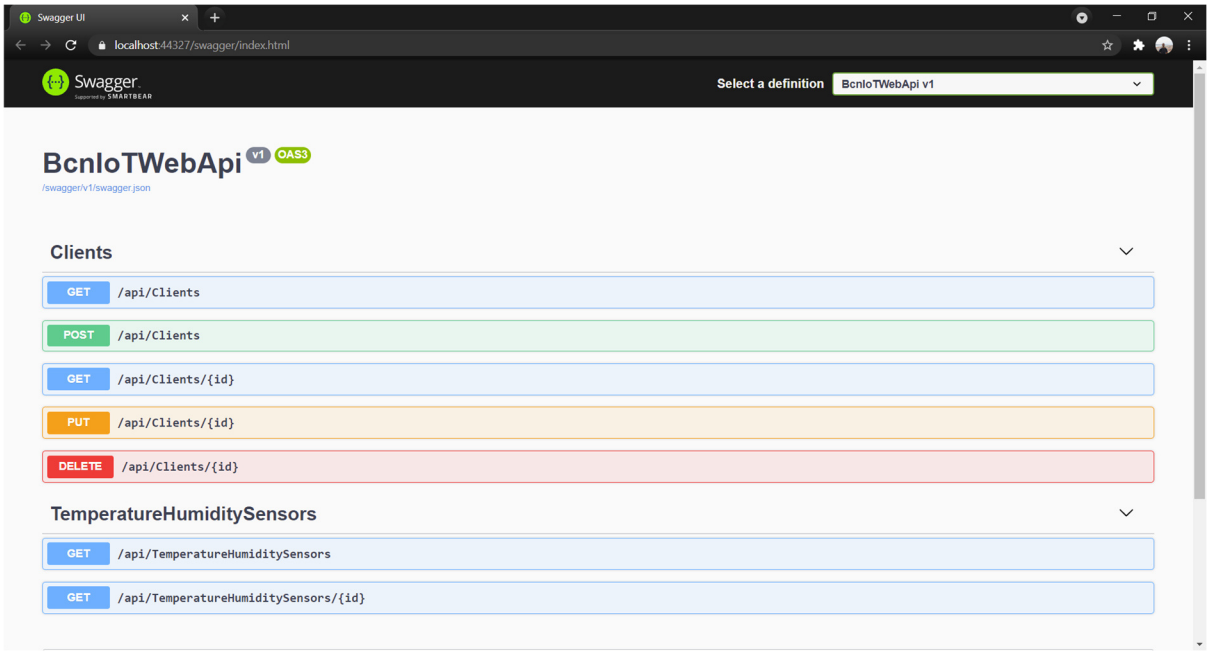


Figure 48. Swagger UI in localhost environment

5.5.2 Services

Using a generic ComosDB service (ComosDBService.cs), other two services (ClientService.cs and SensorS1Service.cs), this layer mainly gets data from the document database in Azure and serve it to the controller.

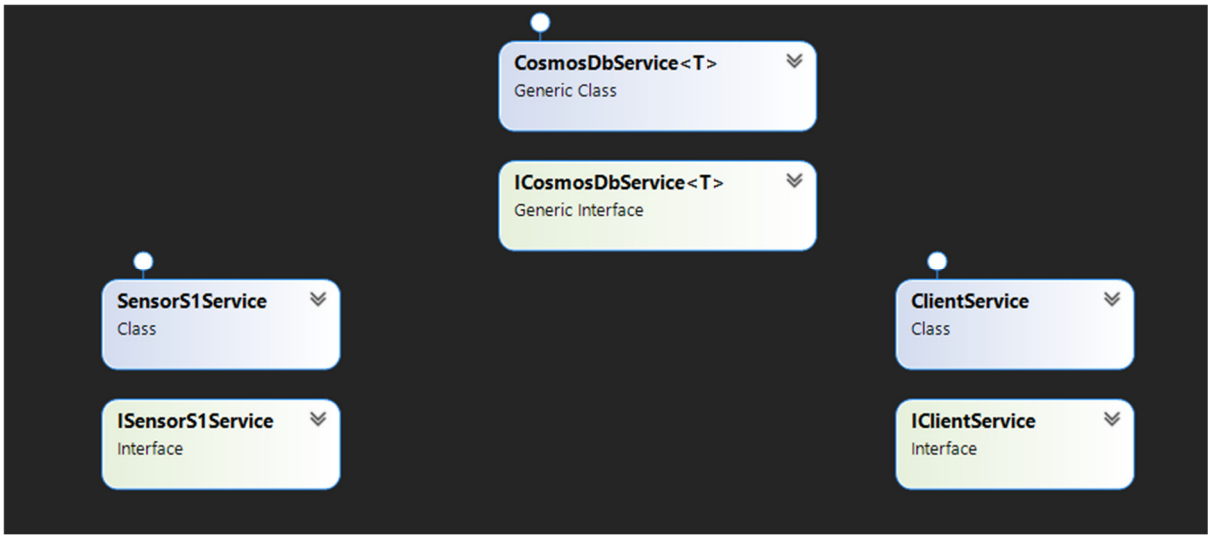


Figure 49. WebAPI. Services classes and interfaces

5.5.3 Models

This project contains only classes for data modelling, used by the other layers.

A dependency injection design pattern is used throughout the solution to allow easy testing, flexibility and to help having a loosely couple architecture.

Code repository is available in GitHub:

https://github.com/javierporta/bcn-iot-webapi

5.6 Event Trigger

An Azure Function plays the role of an event trigger. It receives HTTP requests generated as an output of the Stream Analytics Job on the edge and process them. This process consists in the following steps. In Azure function app is called *PushNotificationFunctionBcnIot* and the function name is *HttpTrigger1*. Workflow is explained in *Figure 50. HttpTrigger1 workflow* and console result logs are shown in *Figure 51. Example of the console output of the HttpTrigger1 function. First with a max threshold of 30ºC and then changing to 27ºC to trigger push notification.*

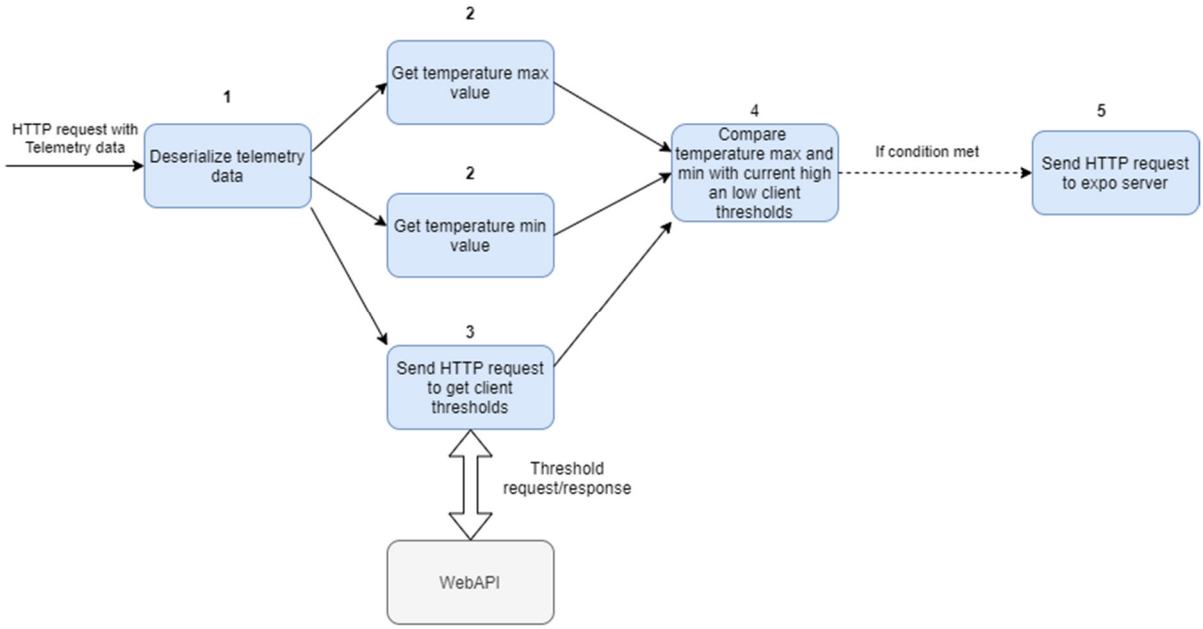


Figure 50. HttpTrigger1 workflow

1. Deserialize request sent by the Stream Analytics Job with current temperatures. This last one, may gather many samples of telemetry and send all as an array. For example, three samples of three different seconds (or from different sensors), until another request is sent.
2. From those samples of current telemetry data, get the maximum and minimum temperature.
3. Make a request to the WebApi, which get current user temperature thresholds from CosmosDB database.
4. Compare lowest and highest temperature with the high and low threshold.

5. If highest temperature is greater than high temperature threshold or lowest temperature is lower than low temperature threshold then sends an HTTP request to Expo Server (push notification server)

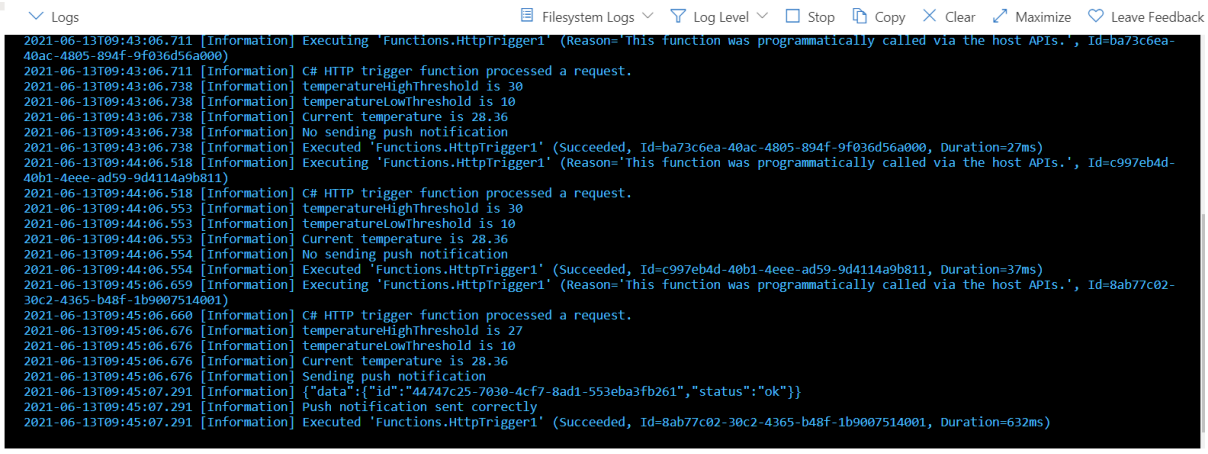


Figure 51. Example of the console output of the HttpTrigger1 function. First with a max threshold of 30ºC and then changing to 27ºC to trigger push notification.

Azure function has been programmed using C# and its code can be viewed using the following link.

https://github.com/javierporta/TriggerHttpFunction

5.7 Push Notification Server

As a way of taking advantage of the tools used, and because it is only required to send and receive simple push notifications an Expo Push Notification server is used to send push notification to the mobile application. By this way, all the hassle with native device information and communicating with APNs (Apple Push Notification service) or FCM (Firebase Cloud Messaging) is taken care of behind the scenes [26].

Once the application being configured and programmed to receive push notifications using expo server, it is only needed to create a POST HTTP request to the push notification server with a unique token in order to send a push notification to the application. It is worth to mention that no authentication was implanted here, neither a filter by client, but both are possible to do.

[26] https://docs.expo.io/push-notifications/overview/

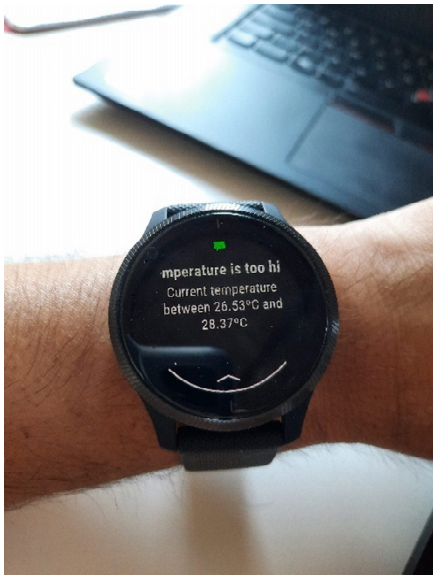


Figure 52. Push notification arriving to a Garmin Venu smartwatch connected to the smartphone that has installed the app.

5.8 Mobile Application

A React Native[27] application written in Typescript was developed in order to present in a fancy way current telemetry data of the sensors to final users. This cross-platform tool allows us to reuse the same code to deploy a web application, an Android native application and an iOS native application. This tool has been used together with Expo[28], a framework and a platform for universal React applications which consists in a set of tools and services built around React Native and native platforms that help to develop, build, deploy, and quickly iterate on iOS, Android, and web apps from the same JavaScript/TypeScript codebase. Among those tools, the push notification workflow used in this project and real time debugger using emulators or real devices.

The application basically communicates with the Web API, sending HTTP request to get telemetry data stored in the ComosDB database and showing it to the client. It also allows users to change its welcome name and high and low temperature threshold, apart from receiving push notification.

That given said, the react native application contains 3 tabs.

1. **Tab 1**. List of sensors by client, showing the MAC of each of them. When user tap one, it accesses to the last measured value of temperature and humidity, along with the date it was taken.

[27] https://reactnative.dev/
[28] https://expo.io/

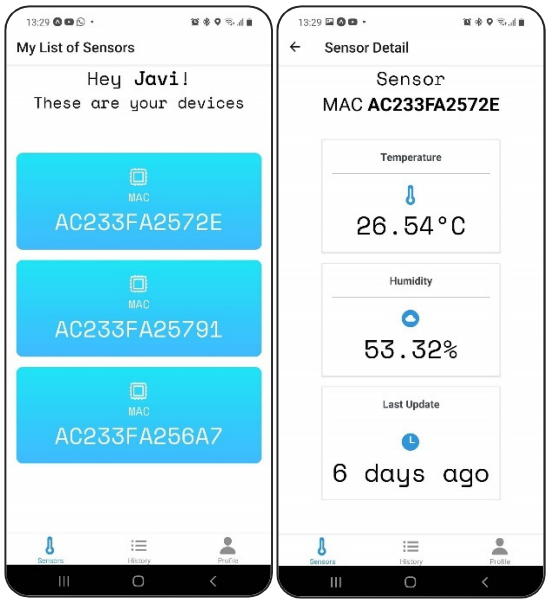


Figure 53. Mobile application. Tab 1: Sensors list and sensor detail.

2. **Tab 2**. History of last 100 measured values. Implemented pull down to refresh the list.

Figure 54. Mobile Application. Tab 2: Sensors' history

3. **Tab 3**. Client's data. Showing name and threshold of the users and allowing to update those values.

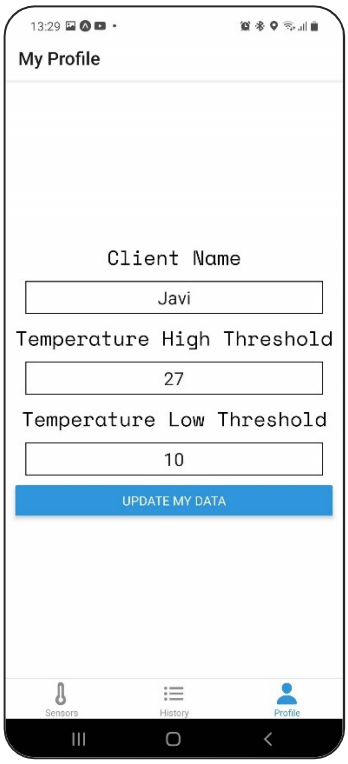


Figure 55. Mobile Application. Tab 3: My profile form

React application code is stored in the following Github repository.

https://github.com/javierporta/bcn-iot-mobile-app

5.9 Streaming Analytics

Following the Microsoft environment, a serverless real-time analytics named Azure Stream Analytics[29] is used as the streaming analytics tool. This tool is used to create an end-to-end analytics pipeline that is effortless production-ready. Given pipeline is configured to get telemetry data of IoT Hub, run a query using familiar SQL syntax and generates an output what can be used as an input of other tools. This powerful tool is capable of analyse millions of events at sub second latencies.

Three steps are needed to configure this tool. An input, what is the source of data, the output, where it should send the results and a query, how it should process input data to convert it in the output data, including select and filtering operation for instance (Figure 56. Azure Stream Analytics dashboard).

[29] https://azure.microsoft.com/en-us/services/stream-analytics/

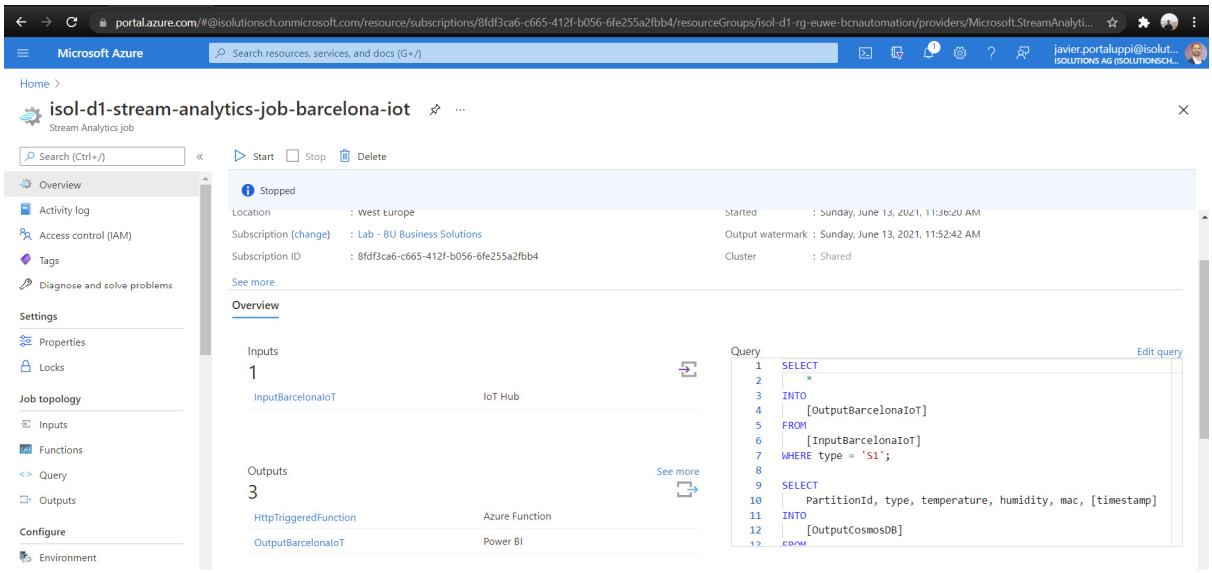
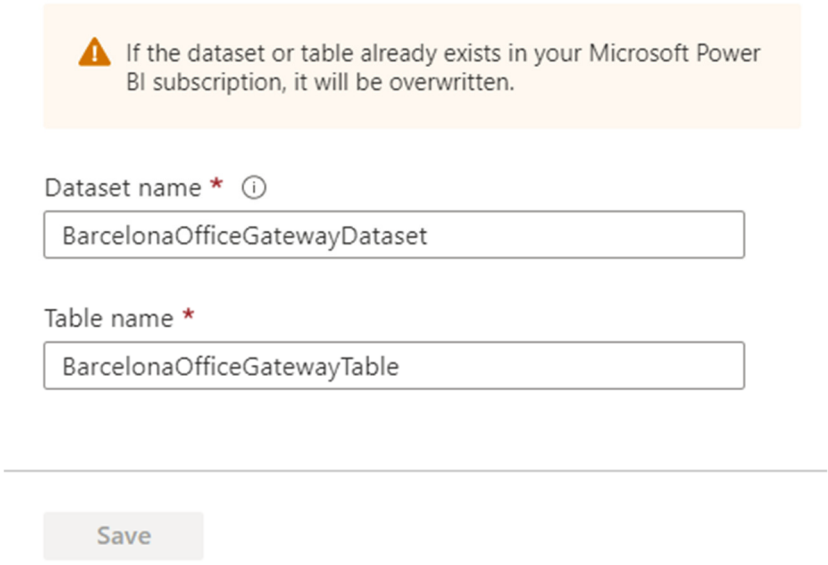


Figure 56. Azure Stream Analytics dashboard

The IoT hub instance, which is in the same Azure resource, has been set as the input of the job. A power BI account has been connected to the job and a dataset and a table inside it has been linked as an output. Also, a ComosDB database and an Azure function has been added as an output of this job.



Finally, the query only places the data of the IoT hub to the Power BI table just filtering the gateway packets by its type. It's only taken into account the "S1" type that makes reference to S1 devices temperature and humidity telemetry data, and the result can be visualized in real time, as it is shown in Figure 57. Azure Stream Analytic Job. Query.

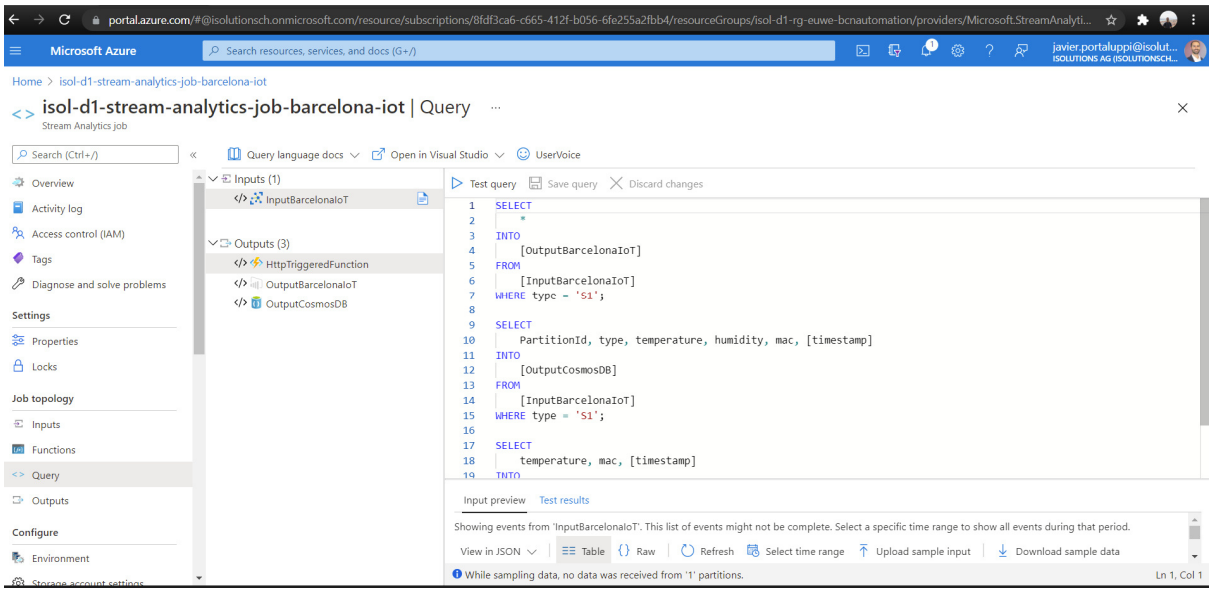


Figure 57. Azure Stream Analytic Job. Query

5.10 Web Report

In order to show sensors data in a nice and interactive way from any browser, a Power BI[30] report was created. Power BI is a Microsoft tool that unifies data from many sources to create interactive, immersive dashboards and reports that provide actionable insights and drive business results. As being part of Microsoft, it integrates and works really well along the Microsoft environment, for instance other Azure platforms.

Web report uses hot path flow to show sensors data instantly, meaning that it gets the data from streaming analytics query, as previously mentioned.

Using the dataset filled by the stream analytics job a report was created and shared between al the member of the company (Figure 58. Power BI web report). It contains, for each of the three sensors connected, a line chart showing temperature and humidity values during the last four weeks, current battery level, its position, its MAC, average, maximum and minimum of temperature and humidity. One of the advantages of this type of report is that it is interactive and final users can also apply filters, add groupings, expand charts and more, to dive in some more details according to their needs.

[30] https://powerbi.microsoft.com/en-us/

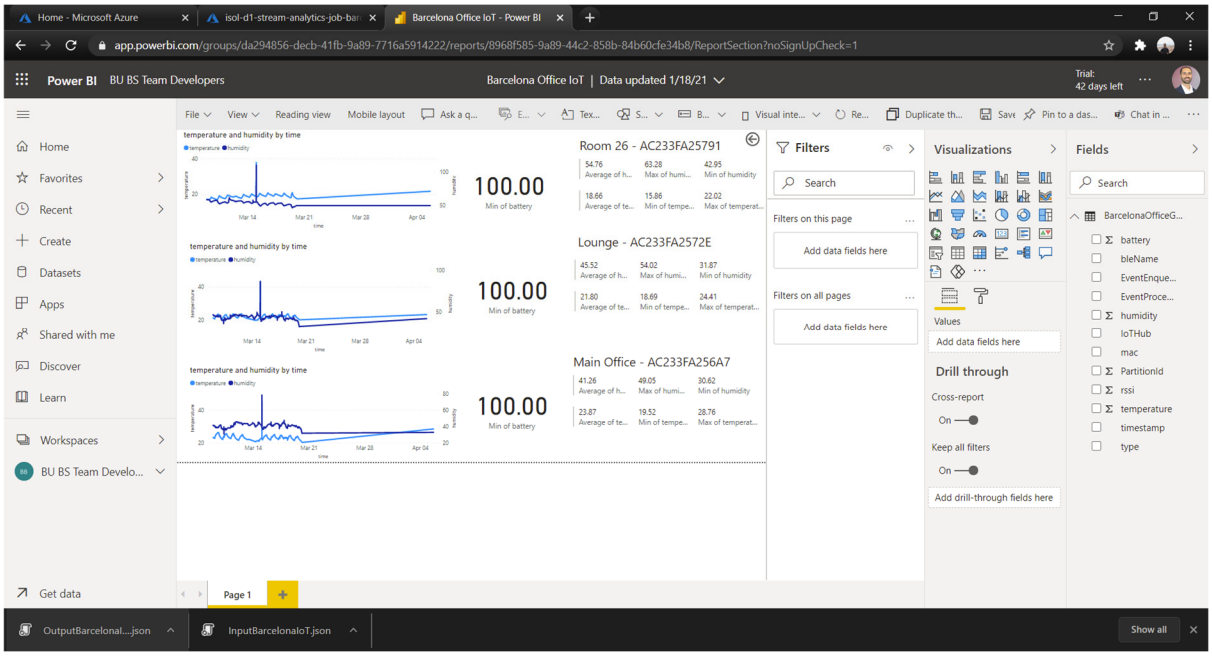


Figure 58. Power BI web report

In a real-world scenario, web report should use a cold path for reducing cost and because of the type of output it is (final users will not have this report opened all the time), but for testing tools capabilities it has been used in the cold path, connected with the azure stream analytics job.

6 Tests and Results

Next, to be described the common nowadays challenges in any IoT solution which, indeed, apply to this project.

Connectivity

Enabling a seamless flow of information to and from a device, infrastructure, cloud and applications, is a top IoT challenge because wireless connectivity is highly complex, and dense device deployments further complicate operations. Yet, mission critical IoT devices are expected to work reliably without fail even in the toughest environments. Fast-evolving wireless standards add to the complexity, and engineers face constant challenges in keeping pace with the latest technologies while ensuring devices can work seamlessly throughout the ecosystem. (Wong, 2020)

Continuity

Ensuring and extending battery life, one of the most important considerations for IoT devices. A long battery life is a huge competitive advantage in consumer IoT devices. For industrial IoT devices, a battery life of five or ten years is the common expectation. For medical devices such as pacemakers, device life can mean the difference between life and death. And of course, battery failure is not an option. (Wong, 2020)

Keeping IoT hardware updated

It is critical to calibrate IoT sensors on a regular basis. Next-gen sensors are embedded in many different devices, including panel meters, chart recorders, current clamps, power monitors and more, and it is difficult to synchronize the dataflow between all this hardware without the help of a professional team. (Matthews, 2018)

Privacy and security

As the IoT become a key element of the Future Internet and the usage of the Internet of Things for large-scale, partially mission-critical systems creates the need to address trust and security functions adequately. New challenges identified for privacy, trust and reliability are:

• Providing trust and quality of information in shared information models to enable re-use across many applications.

• Providing secure exchange of data between IoT devices and consumers of their information.

• Providing protection mechanisms for vulnerable devices. (Keyur K Patel, 2016)

Coexistence

With billions of devices, congestion in the radio channels is a problem that will only get worse. To address wireless congestion, standards bodies have developed test methodologies to evaluate device operations in the presence of other signals. When radio formats do not detect each other, collisions and data losses will occur. (Wong, 2020)

Costs vs usability

IoT uses technology to connect physical objects to the Internet. For IoT adoption to grow, the cost of components that are needed to support capabilities such as sensing, tracking and control mechanisms need to be relatively inexpensive in the coming years.
(Keyur K Patel, 2016)

This project was a continuous trail error process for each of the modules involved until the solution presented was achieved. Individual results for those modules were showed in previous section. Nevertheless, all of them play a fundamental role and their synergy translates into an integrate and scalable platform for measuring values of a physical place, in this case, smart office, and showing this information on real time to final users, office general managers, in order for them to take immediate actions and benefits from providing a great place to work for all employees. Even though solution is tied to iSolutions Barcelona office at the moment, it is easy to add new tenants, and even easier to add new BLE sensors of any type. Next, each module is going to be analysed.

Minew S1 Sensors and Minew application. Temperature and humidity sensors responded very well during all the time they have been used. All of them have consistent values when comparing them with themselves. Battery consumption is quite low when is well configured, after 6 month of everyday use the three of them have 100% of battery level. Also, there was no connection problem having one of them in other office, about 15 meters away, and power can still be increased to reach longer distances. Price is reasonable but there are way cheaper temperature sensors if only basic functions are needed. Last but not least, mobile application developed for the same manufacturer is intuitive but is still needed to read the manual for some configuration parameters. This really makes each sensor really easy to configure. For security reasons there is a hardware button on the sensor that enables and disables connectable mode, which is a great advantage over cheaper competitors.

Minew G1 Gateway. Performance of this device was quite reliable. Configuring it was not an easy task, since reading the brief manual was not enough to know everything was needed. At the time project was performed there was no native connection with cloud

providers, so extra configuration for using HTTP or MQTT to connect to the cloud is needed. It offers filtering BLE packages by MAC, names and types of packages by using regular expressions. It comes with the ability of opening and closing access point connections, by holding a button for two seconds, but if you do not pay attention and hold it longer you do a factory reset, which will end up losing all the configuration. There is no way of exporting/importing the current configuration. The newer version can store data in a SD card in the case there is not network connection, which helps to retain data for further analysis. Also, it has a considerable large range, with 300 meters maximum in open space for BLE and 90 meters maximum in open space for WiFi, there was no problem at all with the dimension of the two rooms of iSolutions office. With those values if we had a place to cover of 3 floors of 850m2 we would only need one (or at most two) gateways depending on the walls, their thickness and other devices that may interfere in the signal. There is no official limitation regarding the quantity of devices that can be connected to the gateway but may be guessed doing a stress test with hundreds of devices at the same time.

Azure IoT Hub. This platform was not very well documented. To use a secure connection with MQTT it was needed to research on blogs instead of official documentation. Probably, Azure IoT Central is easier to set up. However, once is configured it worked nicely. It allows to add different sensors and different gateways and redirect or trigger other services within the Azure ecosystem. It is meant to be connected with Azure Stream Job or Event Hub which are paid services of the platform. It also provides different SDK for different languages to emulate devices and send/receive packets from/to the cloud. Furthermore, it allows the connection of up to 1 million devices[31] with the possibility to ask for more capacity contacting Microsoft support. This is a very important detail since it opens the door to scale current solution to a multitenant solution.

ComosDB database. It was a real success to use it. Easy to deploy and to use. SQL knowledge is needed for querying. As it is no relational, there was no need to deal with tables relationships and the best is that different documents with different data can be used. This mean that, for instance, if we add a new sensor and want to store its values, there is no need to make any change to the database structure, helping to the scalability of the solution. As a downside, it might be costly when data becomes huge.

.Net Core REST Web API hosted in App Service. Coming from a C# and .Net background this was a clear win too. There is a .Net library as a wrapper of ComosDB which make communication simple. Easy to develop and publish to the cloud directly from Visual

[31] https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-quotas-throttling#other-limits

Studio, using integrated Azure tools. A free tier app plan was used, so no costs involved here.

Azure Function using HttpTrigger. This was an unexpected unplanned tool which ended up being the brain of the real time push notification system. Azure has been pushing the usage of Azure Functions a lot and improving it through the last years. It can be developed using multiple languages and it is totally flexible, as you can write custom code according to your needs. It integrates really well being an output of a Streaming Analytics Job. A different Azure subscription was used and worked the same way, using the manual configuration, so it is definitely multi-tenant friendly. It has a live console to debug requests on real time. Cost may be regulated by consumption, providing elasticity for high and low demand, or fixed depending, on the resources beforehand planned to be used (less elastic but safer)

React Native application using Expo. If you come from the React world, the learning curve is quite steep, since same engine is being used and only it is needed to learn the new components specific for cross platform. Making one app and deploying to the web, Android and iOS is just amazing. However, it is true that might start to become tricky when it comes to use more specific smartphone sensors or functionalities, which in this simple app they have not. Using Expo with it eased developing and debugging in different platforms. Also, implementing push notification using expo server has made things way much simpler than going for a native approach. Highly recommended for small and simple applications.

Streaming Analytics Job. This is, definitely, the core of the whole solution. It takes telemetry data and send it to three different sources. If it fails, the rest of the modules will not receive any data. It integrates perfectly with other Azure services and tools. It is the key to analyse information on real time, since other tools provided by Azure may last some seconds, event minutes. The big downside is that it is quite expensive for what it does.

Power BI Report. The web report tool is intuitive and easy to use. Building a report can be a work for just a couple of days and you do not need to know programming to do so, so anyone can build it. It is easy to share it among people of the same organization or outside it. As mentioned before, it is true that there is no certain need to use it connected as an output of the streaming analytics job (hot path). It can be built using the CosmoDB database as a source, opening the door to plenty more of report visualizations, taking into account stored data. For example, highest and lowest temperature in January, average temperature in office A in year 2020, time zone with maximum temperature, temperature variation between different sensors placed in different offices, and so on.

7 Conclusions

Providing a great place to work is a challenge that nowadays companies must deal with if they want to keep the people on the organisation and attract new talents. One way of helping companies' managers to keep pleasant offices is measuring values in the workplace that can affect employee's mood and productivity, such as temperature, humidity, luminosity, noise, CO2 level, capacity, among others. To achieve that goal, a set of devices, tools and software integrated in a IoT ecosystem can help to measure those values to further take some actions, providing workers a better place to work and keeping them happy, boosting in that way their productivity which is benefit for both employers and employees. It is also worth to mention that there is a benefit to the planet too, since this aims to help with the optimization of the usage of the air conditioning and the heater.

Even though working in IoT is always a challenge since it requires dealing with interoperability, high demand, availability, energy consumption, costs, different manufacturers, among others, this work has reached its objective of developing an IoT solution that allows companies managers to measure humidity and temperature values and proved that it is totally scalable to multiple sort of sensors and different tenants.

This work is not the end of a project but the start of it. It is a proof of concept that demonstrate that the Internet of Things can be used in the offices to improve operation and what was raised at the very beginning is doable. That means that, there is a lot of room for improvement and adding features to it as a future work. For example, adding luminosity, noise an CO2 level sensors, adding authentication to the mobile application and web API to finish the multitenant flow, refactoring the mobile application (creating a service layer and new reusable components). Also, to make the application available for employees and create a section where they can evaluate how happy are they with current measured values (temperature, noise, etc.) at any time, as a way to rate employee satisfaction. Furthermore, temperature sensors can be connected to the temperature system of the office so it can be automatically regulated regardless human intervention. Another interesting idea would be to create dynamic reports in the mobile application crossing data between office measured values and employee's productivity. Other good challenges may be to try other cloud providers, technologies, devices or tools to try to reduce costs and deploy them to production environments.

Regarding the company this project was focus and developed in, iSolutions Barcelona, the Power BI report has been shared with all employees and everyone is happy to see current temperature and humidity values of the offices.

Internet of Things already offers a bunch of effortless applications and usages, but the trend is that it will be continuing offering more and more. Devices of different manufacturers are each time easier to integrate with each other and also with cloud providers. There are a lot of cloud services that are ready to use with a high potential and a minimal cost. Tools are there, available to all of us, and it is up to us to create the best of them to make our lives easier while taking care of the planet at the same time.

8 References

AMQP. (2021, 03 06). *AMQP is the Internet Protocol for Business Messaging*. Retrieved from AMQP: https://www.amqp.org/about/what

André Glória, F. C. (2017). *Design and implementation of an IoT gatewayto create smart environments.* Lisbon, Portugal: The 8th International Conference on Ambient Systems, Networks and Technologies.

Ashraf A. Shikdar, B. D. (February 1995). *A field study of worker productivity improvements.* Elsevier.

Bellet, C. a.-E. ((October 14, 2019)). *Does Employee Happiness have an Impact on Productivity?* Saïd Business School .

HiveMQ. (2019, 07 17). *Client, Broker / Server and Connection Establishment - MQTT Essentials: Part 3*. Retrieved from HiveMQ: https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/

Holst, A. (2021, 01 20). *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030*. Retrieved from www.statista.com: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/#:~:text=The%20number%20of%20Internet%20of,China%20with%203.17%20billion%20devices.

Jaycon Systems. (2017, September 28). *Bluetooth Technology: What Has Changed Over The Years*. Retrieved from Medium.com: https://medium.com/jaycon-systems/bluetooth-technology-what-has-changed-over-the-years-385da7ec7154

Josie Hughes, J. Y. (2015). Development of wireless sensor network using Bluetooth Low Energy (BLE) for construction noise monitoring. *International Journal on Smart Sensing and Intelligent Systems*.

Kevin Townsend, C. C. (2021, February 3). *Getting Started with Bluetooth Low Energy*. Retrieved from Oreilly: https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html

Keyur K Patel, S. M. (2016). *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges.* Vadodara, Gujarat, India: International Journal of Engineering Science and Computing.

Matthews, K. (2018, April 20). *5 big challenges still facing the Internet of Things (IoT)*. Retrieved from Big Data Made Simple: https://bigdata-madesimple.com/5-challenges-still-facing-the-internet-of-things-iot/

Microsoft. (2021, 02 20). *Azure IoT - Internet of Things platform*. Retrieved from Azure Microsoft: https://azure.microsoft.com/en-us/overview/iot/

MQTT. (2021, 02 20). *MQTT - The standard for IoT messaging*. Retrieved from MQTT: https://mqtt.org/

OASIS. (2019, 3 7). *MQTT Version 5.0*. Retrieved from OASIS: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

Open Automation Software. (2021, 02 14). *What is an IoT Gateway?* Retrieved from Open Automation Software: https://openautomationsoftware.com/open-automation-systems-blog/what-is-an-iot-gateway

Padmapriya R, M. J. (2017). *IoT-Based Smart Office System Architecture Using Smartphones and Smart Wears with MQTT and Razberry.* ICDECT .

Pieter Bonte, F. O. (n.d.). *User-Friendly and Scalable Platform for the Design of Intelligent IoT Services: a Smart Office Use Case.* Gent, Belgium: Ghent University.

Priyadarshiny, U. (2020, 11 25). *Real World IoT Applications in Different Domains*. Retrieved from Edureka: https://www.edureka.co/blog/iot-applications/

RedHat. (2021, 02 20). *What are cloud services?* Retrieved from RedHat: https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services

Senion. (2019, 09 13). *What is a Smart Office?* Retrieved from Senion: https://senion.com/insights/what-is-a-smart-office/

T F Prasetyo, D. Z. (2018). *Prototype of smart office system using based.* IOP Publishing Ltd.

WebGeometrics. (2021, 02 21). *Know how GCM push notifications works on android devices*. Retrieved from WebGeometrics: https://www.webgeometrics.com/know-how-gcm-push-notifications-works-on-android-devices/

Wong, S. H. (2020, June 30). *The Top Five Challenges of IoT*. Retrieved from IoT Evolution: https://www.iotevolutionworld.com/iot/articles/445866-top-five-challenges-iot.htm