2015-07

Development of an IOT front-end with the Fiware platform for Smart City solutions

González Mallo, Andrea

http://hdl.handle.net/10016/23824

CARLOS III DE MADRID UNIVERSITY BACHELOR DEGREE IN COMMUNICATION SYSTEM ENGINEERING

BACHELOR THESIS

Development of an IOT front-end with the FIWARE platform for Smart City solutions

A thesis submitted by Andrea Gonzalez Mallo under the tutorship of the Professor Victor P. Gil Jiménez and the mentor Marcos Reyes Ureña, CTO in Industrial IoT / FIWARE at Telefónica I+D S.A.

Madrid
July 2015

Acknowledgements

Con este proyecto, pongo fin a los últimos seis años de mi vida, cuatro de ellos en la Universidad Carlos III de Madrid. Durante todo éste tiempo, en el que he vivido muy buenos momentos gracias a la universidad, pero alguno que otro malo, mis padres han estado siempre ahí para apoyarme no solo en lo económico si no dándome fuerza y animándome a continuar. Papá, Mamá, todo lo que habéis hecho por mí no se puede agradecer solo mencionándoos aquí, porque si he llegado precisamente hasta aquí ha sido gracias a vosotros, muchísimas gracias por haber confiado en mí y creeros esto más que yo. Os quiero y espero que estéis siempre orgullosos de mi. Una de las mejores cosas que me llevo de la universidad son los amigos. A los de Oviedo, que parece que lleváis ahí toda la vida, y espero que sigáis estándolo. Ángel, Cris, Diego, Ana, Adri, sois *una xentiquina perbuena, y débovos folixa asgaya a la fin d'esti proyectu.* A los dos sufridores por excelencia de éste proyecto: la que se ha convertido en mi hermana de Madrid, Ana, muchas gracias por la paciencia, porque haber vivido conmigo éstas últimas semanas tiene muuucho mérito; y a ti, que eres quien más me ha aguantado sin duda éstas semanas, hasta el último momento. Intento agradecértelo todos los días, porque ya sabes que creo que sin tí no habría acabado en este curso, éste proyecto es un poco tuyo también. A todos, ahora empiezan los planes y las aventuras!

También quiero mencionar a mis compañeros de Telefónica, en especial a Manu y Xavi por haberme dado la oportunidad de trabajar con ellos. Habéis creído en mí más que yo misma, y hasta ahora habéis tenido razón... es increíble todo lo que he aprendido gracias a vosotros. A Marcos, por las mismas razones y por introducirme en el universo del IoT, que es la raíz de mi trabajo.

Por último, el agradecimiento más grande: a mi hermana, por ser la que mejor me entiende y apoyarme aunque sepa que me estoy equivocando (no sin recordármelo). Aunque seas más pequeña, eres toda una grande!

Resumen

Internet de las Cosas es un tema de actualidad en el mundo de la tecnología. Poco a poco, el IoT se va convirtiendo en una necesidad para la población, y en una mina de oro para inversores y empresas. Por ello, ofrecer ofrecer una aplicación exitosa es el objetivo de muchas empresas, que quieren despuntar en campos como la telemedicina, la domótica, el Internet de las Cosas industrial, o las Smart Cities. Dentro de este marco, pueden encontrarse colaboraciones entre empresas de la talla de Telefonica I+D, Orange, Thales, Siemens, o IBM, que han contribuido junto con la Unión Europea en el desarrollo de una plataforma de IoT que busca integrar un entorno planificado y común para todos los componentes y clientes de la misma. Por la versatilidad y adaptabilidad de la plataforma, es altamente indicada para proyectos a gran escala, como el diseño de Smart Cities. El proyecto que aquí se detalla tiene como objetivo demostrar las capacidades que caracterizan a la plataforma, utilizando el area de aplicación que mejor pueden exponerlas: el terreno de las Smart Cities. En primer lugar, se analizarán otras plataformas ya comercializadas de Internet de las Cosas. En segundo lugar, se investigarán otros modelos punteros de Smart City ya existentes. Finalmente, un modelo simulado de Smart City que satisfaga servicios haciendo uso de la plataforma será llevado a cabo. Utilizando datos reales de la ciudad de San Francisco, se ha modelado un sistema inteligente para satisfacer algunas de las necesidades de sus ciudadanos, de forma simulada. Con la finalidad de probar la versatilidad de la plataforma, se llevarán los datos publicados en ella a una herramienta online de visualización ajena a la misma. Por otro lado, para probar la fiabilidad de la plataforma, ésta ha sido sometida a tests de integración sobre los componentes, asegurando que éstos pueden comunicarse e interactuar. El resultado de ámbos ejercicios se encuentra documentado también en éste proyecto, que concluye con una revisión positiva de la plataforma.

Abstract

Internet of Things is a cutting edge topic in technology nowadays. Little by little, IoT is becoming a need for people and a goldmine for research companies and investors, who aspire to provide a successful application in fields as eHealth, home automation, industrial IoT or Smart Cities. Inside this framework, companies as big as Telefonica I+D, Orange, Thales, Siemens and IBM, among others, have developed an IoT platform in cooperation with the European Union that looks for a solution to offer an common and organized environment for all the constituents of the platform, and all its clients. Due to the scalability and adaptability of the platform, it is particularly suitable for large-scale projects, such as the design and implementation of Smart Cities. This project attempts to be a demo capable of presenting the platform features, using the area of application that better exposes all its capabilities: a Smart City solution. Firstly, analysing other commercialised IoT platforms, more oriented to offer home automation solutions. Secondly, researching Smart City models already settled worldwide. Finally, a simulated Smart City was developed for this project. Using real data from the city of San Francisco, California, a simulated model of a smart system that satisfies some services based in IoT solutions has been implemented. In order to test the adaptability of the platform, data can be visualized inside a graphical online tool. For testing the reliability of the platform, it was submitted to several integration tests among the components to make sure that they fully can communicate and interact. The result of both testing challenges also figures in this project, that concludes with a positive feedback for the platform.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Economic and social framework

The history of a city would not be possible without its citizens. They are somehow responsible of the location of the city, its design or its growth and development. After technologies were born and raised, cities have evolved as fast as their citizens. Tools and technologies accomplishing the needs of people (or urban systems designed by people) in fields as economy, commerce, industry or culture, have found a complexity conditioned by two factors: first, the amount of available resources to make the technological development work; and second, the strength and knowledge of the human being to accomplish a purpose and take a full advantage of the developed technology.

Thinking about the Industrial Revolution, taking place between the 18th to 19th centuries, a major turning point for every single aspect of human life was marked then. By replacing manual labor by machinery, developing for the first time ever machine tools, and rising a new industrial system known as factory system, every single aspect of the industry changed, in a remarkable way for employment, value of output and capital invested in industry, or investment on modern production methods. As a consequence, the average income and population started to grow massively until late 20th century.

Nowadays, we are facing a new revolution: a Digital Revolution, where the trigger change is replacing analog, mechanical and electronic technology by digital technology. Digital communications have their first footprints inside of an article of the mathematician Claude Shannon, one of the pioneers of digitalization, in 1948. But the first steps in digital communications were not developed until the 60s and 70s with the first packet switched networks, that could transmit messages using different protocols. In fact, 1969 is the year when the concept of Internet was first introduced, after the linkage of several packed switched networks together to transmit shared information, as known as internetworking. Digital revolution finds its opera prima in communications, where people have gained unprecedented capabilities, highlighting permanent availability or hyper-connectivity, that have been the pillars to a new social dependence of being “always on”, where a device becomes part of a person who would prefer a communication in a virtual environment through the device, than establishing a real communication. Social networks allow us to catch up with friends, as well as chatting applications. Furthermore, tweets, blogs or timelines can connect us with the whole world, in a way that shows how we are a part of it: at the end of the day, everyone can be famous just writing the right thing at the right time on the Internet. And this is too attractive for the consumer. And people only want more. People become *smart*.

The word *smart* has lately gained new meanings for its daily usage: from Smart phones, telephones where the functionalities go further than simply make calls; to Smart TVs, televisions able to connect to applications that allow the user a wider range of amusement, enabling a better customer experience. Smart environment is a modern and complex environment, only possible if powerful networks are holding it behind, such as the Internet, a core engine boosting all the pieces around.

Inside this complex environment, Internet of Things raises as the concept selected to explain the interconnection of things with the Internet, based on the idea of transforming these things into Smart Things. However, the idea of what is Internet of Things is yet to be defined, and many companies are still waiting for it to be more solid in order to start developing and investing. The lack of research about Internet of Things can be seen as an issue and an advantage at the same time: on the one hand, the less they research, the poorer the ideas will be in terms of innovation. On the other hand, small companies can take advantage of it, finding a huge oasis of brand new ideas.

As many other customers, Governments and public organisms also use these media to connect to citizens, becoming more accessible, and also urban policies and council’s initiatives find in digital communication channels the accessibility to reach their targets. The vast majority of schemes designed so far within the idea of a Smart City lead to a remodeled idea of the urban models implemented so far. In order to approach how needy is finding new tools to manage better the city, let’s imagine a reality where many items are interconnected providing daily services: from waking up and finding how congested the road is this morning, and when is the best time to leave home; to receive a text message from your fridge saying that you have to buy milk. In order to offer a complex environment with many variables interconnected (such as sensors linked to servers, linked at the same time to a telephone or a web site), it is essential to design a structure suitable to contain all these variables, in the lighter possible way. Cloud-based solutions with a virtually built platform clustered into small components, are nowadays the best approach to display an Internet of Things environment: cloud can be accessed from many sources, and clustered components can be accessed faster if the input point is not common. The truth is that the Smart city has many players in this game, and they must be fairly orchestrated in order to perform a good game.

A Smart city solution is, using another words, an approach of a real city where many physical objects (sensors) connected to the Internet through several telecommunication technologies, are used to enhance the quality of life in the city, by highlighting hidden aspects of the city that are at the same time influential for their citizens whose life would become simpler, and whose behavior would change as well. It is not a coincidence that cities such as Copenhagen, Dubai, Amsterdam, Barcelona or Sidney, who have embraced the Internet of Things and have adopted the first steps into a Smart city model of their cities ecosystem, always rank high on happiness indexes.

Sustainability is another key concept within the scope of a Smart city: the population is growing fast, and cities are facing exponentially increasing urbanization rates: it is estimated that in the next 40 years, the percentage of population living in urban areas will augment in a 72%. In the meantime, the resources required to manage a city generate a huge amount of energetic waste, with side effects as the lack of natural resources, and the environmental issues caused in many natural spaces. In fact, despite nowadays the cities cover a 2% of The Earth, between the 60% and the 80% of global pollution is caused by them, and also a 70% of the Greenhouse Gas Emissions. To reduce the impact of the city in the environment resources it is important to promote the efficient and intelligent deployment of technology and integrate infrastructures, but the question is not only how the energy should be obtained, but what can be done to optimize this energy.

Smart grids, as the digitalization of the traditional grid that composes the electrical network in charge of bringing energy from the power plant to our homes, represent an unprecedented opportunity to move the energy industry into a new era of reliability, availability and efficiency that will contribute to our economic and environmental health: traditional grids produce nowadays more than a million megawatts of generating capacity, connected to more than 300.000 miles of transmission lines. Although the electric grid was an engineering marvel, it is now overloaded, and it is time to move forward. Smart grids are not yet a reality, since it has not been standardized yet, and technology improvements haven't reached their highest peak. Nevertheless, the benefits associated with the Smart grid include a more efficient transmission of electricity, quicker restoration of electricity after failures, reduced peak demand, reduced operation and management costs, improved security or accessibility to the consumer, bringing him the information and tools to make choices about the energy use.

Luckily, there is no need to wait until smart grids are fully available to start thinking how to develop a greener city through a smarter management of resources. In fact, many researchers agree with the idea of refactoring current cities in order to start obtaining benefits, only by investing in public transportation, deploying high-efficiency, or install solar panels on top of buildings to save, by 2050, around $17 trillion, which is a massive amount of money saved, although the most important earning will be saving the quality of life of the citizens, that will find a better urban environment and wellbeing in a Smart Sustainable City defined as *"an innovative city that uses Information and Communication Technologies (ICTs) and other means to improve quality of life, efficiency of urban operation and services, and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social, and environmental aspects"*, according to a definition of the Focus Group on Smart Sustainable Cities, from ITU-T.

And it is not only about the money that digital services will help affording. Smart Cities fulfill many conditions to be the target for many investors, who aim to earn money with them. Smart Cities come implicitly with new vias to offer employement, as new job opportunities, profiles and new working positions arise within Smart City projects in fields such as energy, education, building industry, or informatics and electronics, of course. Talking about numbers, AMETIC (*Asociación de Empresas de la Electrónica, TIC y Contenidos Digitales*) ensures 300.000 new job positions related to Smart City development, before 2017 in our country. Renovables Foundation goes further talking about 500.000 new job positions before 2020 only regarding to ICT projects designed to rehabilitate old buildings that need to improve their energetic efficiency.

Official organisms are in fact starting to invest on smart city research and innovation, and developing policies to make cities more sustainable in the view of Horizon 2020 targets. Beyond the usage of ICT projects for better resource management in the cities, other aspects as greener policies or safer public spaces are also included, in order to fulfill the needs of the population.

In July 2012, the European Commission initiated the European Innovation Partnership on Smart Cities and Communities, to speed up the deployment of smart city solutions in the areas of energy, ICT and transport, helped with funds of more than 3000 partners, and it is estimated that the market value for the IoT in the EU is expected to exceed a trillion euros by 2020, of witch smart city urban services will be 400 billion. In order to lead investors and researchers to the same goal the European commission has initiated in March 2015 the creation of the AIOTI (Alliance of Internet of Things Innovation) flagging the intention for common wealth cooperation among stakeholders and actors of Internet of Things. Two months later, in May 2015, after the Digital Single Market adoption, a step further in accelerating development plans for the IoT was footed: with a core value of bringing down barriers to unlock online opportunities, the DSM consolidates initiatives on security and data protections as well as a new approach to the data economy, including free flow of data, allocation of liability, interoperability of data, usability and access. Basically, sources of free data are open to general public, becoming also a source for the Internet of Things, whose target applications will be fed by as much data as possible.

Within our own borders, RECI (Red Española de Ciudades Inteligentes) chases the consolidation of a co-working environment among cities to share policies based in new technologies that enable economical, social and commercial progress through research and knowledge on communication and information technologies (TIC). Currently working in 62 cities, and wide open to new incorporations, RECI technical committees claim a full cooperation of private and public sectors in order to provide a networking space that supports talent, opportunities and life quality in the city. Figure 1.1 exposes the current cities and communities cooperating with the RECI network, and it has been taken from the RECI website itself [9].

Figure 1.1: Smart Cities in Spain powered by RECI association

1.2 Motivation of this project

When thinking on how a Smart city model should be designed, all the solutions lead to vertical approaches to the different city services: for instance, there might be a solution for mobility or traffic management, which would solve specific problems related to its own development field. Despite this is positive, since is indeed solving a problem, it causes several inconveniences: having separate infrastructures that perform separate operations, idle among themselves, does not facilitate the creation of global ecosystems for developing really intelligent applications and services in the city. Plus, individual and personalized applications that solve one single problem are hard to refactor.

This project considers these inconveniences, and takes up the torch of the research and development of Smart Cities worldwide. The motivation behind is showing how a powerful IoT platform can create an environment capable to fulfill the main urban needs of a city.

After an analysis of the different IoT platforms in the market, one will be chosen due to its power and adaptability. Through a small application based on real data, a walk by the tools to find and handle information, and take insights will be exploited, trying to finally make the results visual and attractive. The project can be also seen itself as an approach to the IoT platform selected, showing how easy is handling data published in there. To the readers of this portfolio, the aim is getting them involved so they can be also users of this platform soon!

As a backup motivation regarding to the Sustainable idea of a future Smart City, this project proves how easy is to deploy green and sustainable ideas when the sources are digital: the application here developed may be small application, but it brings fresh ideas proving that a greener city is possible when the city is *Smart*.

1.3 Goals of this project

This project is committed to solve the complexity of transforming a city into a Smart City using a platform already existing in the market that is suitable to offer an IoT environment powerful enough to sustain applications fed by big data sources, like the ones available in a real city. In order to reach a solution, the following points will be analysed:

1. **Choice of a real city to establish a Smart model in it.** Filtering among cities with open data available, a real city will be chosen based on the characteristics of the city and the people living in it.

2. **Determine the key data sources that will drain the Smart city.** Existing models of Smart Cities will be the reference to find the best data to exploit, and also, regulation regarding to the requirements in a Smart City will be taken into account.

3. **Data modeling from data sources.** Logical analysis to the data sources in order to simulate real data published in real time, as the one that would be collected from sensors located through the city.

4. **Choice of an IoT Platform suitable to gather the data.** FIWARE Platform will be chosen in this case, due to the multiple characteristics that make it fully oriented to host Smart City applications.

5. **Handling data to be published into the IoT Platform.** Data must be suitable to be published into the IoT platform, so it is necessary to reshape it before sending it to the platform, so it can be understood at destination.

6. **Publishing data at an external site.** When discussing real applications within a Smart City, many tools and components from different vendors are orchestrated together for common goals. In order to prove the adaptability of the IoT Platform chosen, an scenario where a third party site communicates with the platform will be developed.

7. **Testing the whole application.** Integration tests proving the communication among the components of the IoT platform will be performed to expose the capabilities of the application.

8. **Conclussions.** Finally, several conclusions regarding to the application itself, and also to the impact of an small development project in the Smart City environment will be provided.

Chapter 2

State of the Art

2.1 Regulation

2.1.1 National regulation

In spring 2015, after many claims on the need of standardizing the concept of Smart City for years, AENOR Technical Committee about Smart Cities has published two rules:

- **UNE 178303:** This rule Establishes the requirements for an optimal and sustainable management of the city services. The requirements will be based on the expenses, risks and work performed due to the maintenance of a Smart city, and it will make it easier for developing good services on it. This document calls every local entity in the country, no matter the size or technological richness of the land. Local authorities will have the freedom to establish the inventory detail or the amount of information collected.
- **UNE-ISO 37120:** This rule determines which indicators will evaluate the quality of a Smart city, through aspects such as quality of life, state on development of urban services, etc. This rule, regarding all cities or towns no matter their size or location, has international scope, and it has recently been added inside the Spanish catalogue of technical rules.

The commitee in charge of preparing both rules, AEN/CTN 178, is promoted by the SETSI *(Secretaría del Estado de Telecomunicaciones, y para la Sociedad de la Información del Ministerio de Industria, Energía y Turismo)*, and it is conformed by more than 300 experts, spreaded in 5 subcommitees in charge of: infrastructures, semantics, govern and mobility, energy and environment, and tourism. As future projects, it is remarkable the future rule UNE 178104, regarding Integrated Management Systems over the Smart City.

2.1.2 International regulation outside ISO

ITU-T

ITU-T Study Group 5, responsible of Environment and climate change issues, inside of a Focus Group on Smart Sustainable Cities, aims to act as an open platform for smart-city stakeholders and ICT organizations to exchange knowledge in the interests of identifying the standardization frameworks needed to support the integration of ICT services in Smart Cities.

IEC

Since 2013, the Systems Evaluation Group (SEG) on Smart Cities works to consider a Smart City as a system of systems in which smaller projects inside a Smart City should be reviewed by a Systems Committee to oversee standards in this area.

IEEE

After years of work on the infrastructure, networking, generation automation and distribution necessary to bring new energy to the cities, the results are a large number of active standards and standard projects for Smart Cities in categories such as Smart Grids, eHealth, or IoT.

European Commision

By launching a Smart Cities and Communities European Innovation Partnership (SCC), the European Commision aims to boost the development of smart technologies in cities by encouraging the research in resources that support Smart City solutions. However, standardization has not arrived yet.

ETSI

In collaboration with CEN/CENELEC, a Coordination Group has been set up to cover the area of standardisation work for Smart and Sustainable Cities and Communities, where the SCC has acknowledged the key role in the development of an standard regarding to Smart Cities within the European scope.

ANSI

On April 2013, a Joint Member Forum composed by experts from standards developing organizations, industry and government, where topics related to international Smart City initiatives are discussed.

2.1.3 International regulation inside ISO

ISO/TC 268

TC 268 is in charge of the Sustainable development in communities, focusing on the development of a management system standard. The sub-committee ISO/TC 268 SC1, is dedicated to smart urban infrastructures. Inside of their work scope, the rule 37120 establishes global city indicators for city services and quality of life.

ISO/TC 163 and ISO/TC 205

A joint working group contributes in the coordination among these two areas, whose research fields are *Building environment design* (TC 205) and *Thermal performance and energy use in the built environment* (TC 163).

ISO/TC 257 and ISO/TC 242

On the one hand, 257 Technical Committee plays a key role in cutting global energy consumption, with rules based on energy saving and improved energy efficiency, which are he best ways to reduce GHG emissions and consume less resources. On the other hand, TC 242 focuses on energy management through policies in energy performance and energy supply.

ISO/TC 59

This commission is concerned on civil engineering works. Throughout the subcommittee SC14, they balance possible environmental and economical impacts.

ISO/TC 223

Develops standards social security issues for private and public organizations, in such areas as emergency management or mass evacuation.

ISO/TC 241

Road traffic safety management systems. ISO 39001 for road traffic safety will assist providing a structured and holistic approach to road traffic safety.

ISO/TC 204

Technical committee focused on the standardization of Intelligent Transport Systems (ITS), regarding to information, communication and control systems in any surface transportation.

ISO/TC TMB

Technical Management Board. Within the framework of policies established by the ISO, TMB has the responsibility for the general management of the Technical Committee structure.

2.2 Technology and trends

The future Internet domain landscape agglutinates a wide range of technologies regarding the topics involved in the implementation of Smart City based applications. This chapter contains a brief summary about many trendy technologies that are closely related to Smart Cities development.

Machine to Machine (M2M)

M2M refers to the set of technologies that allow both wireless and wired systems to communicate with other devices of the same type. Technologies included in the M2M specification enables online data gathering, remote control and process automation.

Internet of Things (IoT)

The Internet of Things refers to the interconnection of electronic devices within the existing Internet infrastructure. Typically, IoT is expected to go further than M2M technologies, offering advanced connectivity solutions regarding devices and embedded systems.

Embedded systems

A computer system with a specific function within a large hardware piece is defined as embedded system. Nowadays, many of the devices and sensors available in the market are under this definition. For this reason, big networks of embedded systems (embedded networks) are the physical scenario where many Smart City solutions lay.

Service oriented architecture

Software architecture design based on distinct pieces of software providing applications and services to other applications (as a modular structure). The terminology of service orientation owes its name to the fact of being independent of any vendor, product or technology.

Open Data

If a Smart City is an engine, data is the oil running through it. The availability of data, its use, or quality determine as well the quality or usefulness of a Smart City application. In many cases, the reliability and richness of Smart City solutions depend on adding data from various sources, which is not possible when data is not publicly open.

Big Data

Data can be as complex, shapeless and large as a city, and traditional storage systems cannot follow the rhythm, so they are not accurate to store a vast piece of information always on change. On top of that, Big Data tools offer a higher complexity level allowing the access to insight data, not visible at first sight, that can even unlock new economic sources in the city.

GIS

Geographic Information System, a computer system designed to capture, store, and handle all types of geographical data. In Smart City applications, offering a real time location can be as important as the service information itself. For that reason, GIS applications are essential to analyze spatial information, edit data in maps and present results in a visual way. One of the most popular formats of presenting GIS data is the **shapefile** format. It offers a vectorial representation of a geometrical location and an associated attribute information, combining the two most powerful data sources of information relevant in a Smart City development.

Cloud computing

Cloud computing is the delivery of computing as a service rather than a product, in which software, resources or information are shared and provided to computers or other devices as an utility over the Internet network. It fits so good inside the Internet network concept, that it can be defined as an Internet-based computing, where servers, storage systems and applications are delivered to a grid of computers and devices through the Internet network.

eCommerce

Electronic commerce, or EC, is the buying or selling of goods and services, or the transmission of funds as data over an electronic network, mainly the Internet network. Transactions may occur either business to business, business to consumer (and viceversa) or consummer to consummer.

eGovernment

Electronic Government, e-gov in a shortly name, is a concept directly raised as a consequence of the social influence of Smart Cities. It defines the digital relations between a government and the different sectors of society: government and citizens (G2C), government and business or commerce (G2B), government and employees (G2E) or even government and other governments (G2G). E-government improves broad stakeholders contribution to national development, and improves governance processes at the same time.

eHealth

Electronic Health is the transfer of health resources and health care by electronic means, in three main areas: digital delivery of health information for health proffesionals or health consumers, improvement of public health services using IT solutions, and the usage of e-commerce practices in health systems management.

ITS

Intelligent Transport Systems are advanced applications which aim to provide innovative services realted to differnt models of transport and traffic management, enabling users a safer, more coordinated and smarter use of transport networks. Nowadays, they are a cutting-edge topic in Smart City innovation.

2.3 IoT Alliances and Ecosystems

The Alliance for Internet of Things Innovation comprises many IoT industry players and successful startups, bringing together:

Different industries

Nanoelectronics/semiconductor companies, Telecom companies, Network operators, Platform Providers of IoT and Cloud, and Security and Service providers.

Different sectors

Energy, utilities, automotive, mobility, lighting, buildings, manufacturing, healthcare, supply chains, cities, etc.

International companies

Alcatel, Bosch, Cisco, Hildebrand, IBM, Intel, Landis+Gyr, Nokia, ON Semiconductor, Orange, OSRAM, ABB, ACM, Philips, Samsung, Schneider Electric, Siemens, NXP Semiconductors, STMicroelectronics, Telecom Italia, Telefonica, Telit, Thales, Vodafone, Volvo, Huawei, and many more.

Encouraged by the alliance ring favoured by AIOTI, many of the previous companies have engaged to develop common projects and products, that will be enumerated and explained below.

2.3.1 AllSeen Alliance

- **Leader:** Qualcomm, Software handled by the Linux Foundation.
- **Key members:** Sharp, Haier, LG, Sony, HTC, Silicon Image, Panasonic, Microsoft, Canon, Electrolux.
- **Commercial product:** AllJoyn

AllSeen Alliance is a nonprofit consortium dedicated to driving the widespread adoption of products, systems and services that support the Internet of Everything with an open, universal development framework that is supported by a vibrant ecosystem and thriving technical community. The framework delivers an open platform and common set of service frameworks for allowing things near you (proximal) to share information seamlessly with each other regardless of operating system, platform, device type, transport layer or brand. It consists of an open source SDK and code base of service frameworks that enable such fundamental requirements as discovery, connection management, message routing and security, ensuring interoperability among even the most basic devices and systems. The initial planned set of service frameworks include: device discovery to exchange information and configurations (learning about other nearby devices); onboarding to join the user's network of connected devices; user notifications; a common control panel for creating rich user experiences; and audio streaming for simultaneous playback on multiple speakers. The Alliance hosts and advances an industry-supported open software connectivity and services framework based on the AllJoyn open source project.

Initially developed by Qualcomm Innovation Center (QuIC), Inc (Qualcomm's open source subsidiary), AllJoyn is an open, universal, secure and programmable software connectivity and services framework that enables companies and enterprises to create interoperable products that can discover, connect and interact directly with other AllJoyn-enabled products. AllJoyn is transport-, OS-, platform- and brand-agnostic, enabling the emergence of a broad ecosystem of hardware manufacturers, application developers and enterprises that can create products and services that easily communicate and interact. It is based on messages passed between apps and firmware-based routers. In the AllJoyn scheme, app-to-app messaging is accomplished through routers. This ability to allow direct app-to-app communications eliminates the *middleman* and precludes the need for intermediary servers. Figure 2.1, obtained from a blog post [27] of the IERC (IoT European Research Cluster), enforces the AllJoyn app/router concept, and it exposes the AllJoyn App/Router middleware inserted neatly between the application and workhorse OSI networking layers.

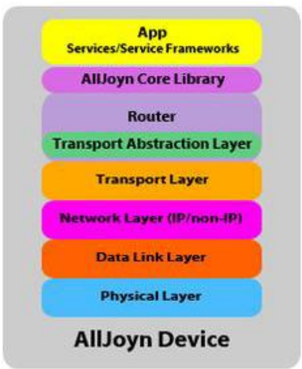

Figure 2.1: AllJoyn device network scheme based on the OSI model

2.3.2 ARM mbed

- **Leader:** ARM, the industry's leading supplier of microprocessor technology.
- **Key members:** ATMEL, NXP, STMicroelectronics, Nordic Semiconductor, Freescale, U-blox, SiliconLabs.
- **Commercial product:** mbed (working on release mbed v3.0)

The Internet of Things (IoT) business unit at ARM is committed to building the technology and ecosystem that is the foundation for creating and deploying low-power, connected and secure IoT devices at scale. They are also building a software ecosystem around these devices to allow true IP-to-the-edge and interoperable IoT applications, concerned on how the Internet of Things will bring the internet to billions of devices and create new ways of seamlessly engaging with technology to facilitate a more connected and secure world. ARM is very strongly positioned at the market (2.5 billion of devices are based in ARM solutions), with a big developers community (around 70.000 developers enrolled).

ARM has created a solution to help lead sustainable growth within the IoT and named it mbed, ARM mbed promotes the idea that the Internet of Things (IoT) is a collaboration between the people who make things, the people who use things and the people who make the services all these things use. The basic hardware of an mbed device is the development board, which includes an ARM microcontroller. The board is capable of small amounts of processing and data storage, as well as communication. There are additional components that can provide even more capabilities. These include sensors, touch-screens, motors, storage extensions and even wheeled robots. The mbed boards run mbed OS - our in-house operating system that offers an easy, consistent way of controlling the hardware and interacting with the cloud. It also offers built-in integration with other ARM tools, such as an automated testing tool. On top of mbed OS sits, software code governs on the hardware components, with the constrain of the hardware's processing time and storage abilities.

The product vision is based on P2P communications mainly using the IEEE 802.14.5 radio, IETF 6LowPAN, IETF CoAP and the main emerging standard for IoT objects/interfaces: OMA Lightweight M2M. After a recent agreement, OMA-LW-M2M will be also supported by IBM's MQTT protocol. mbed supports additional communication alternatives: Cellular, Bluetooth, MQTT, HTTP, etc. Also, a IoT Cloud service is available. It also implements the concept of *Web of the Things*, providing a RESTful architecture. This compendium of services and technologies is represented in the Figure 2.2, obtained from the mbed v3.0 datasheet [17].

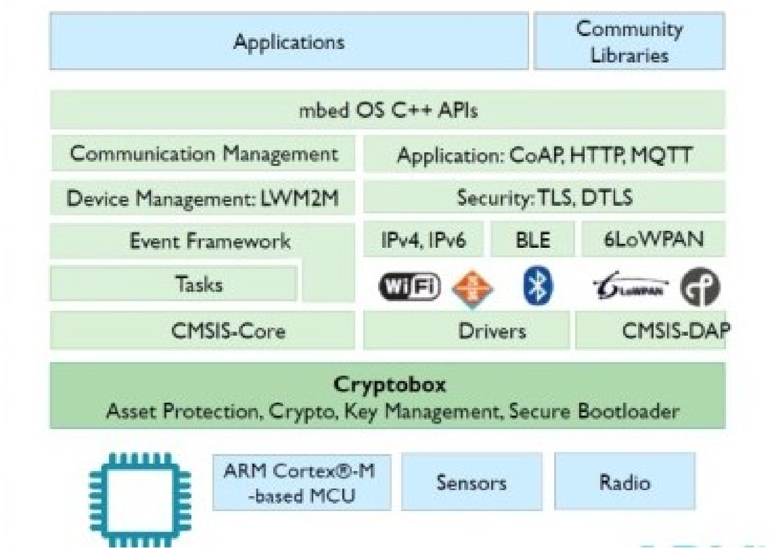

Figure 2.2: Network architecture planned for mbedOS in the v3.0 release, in a high level view

2.3.3 Open Interconnect Consortium

- **Leader:** Intel, the industry's leading supplier of microprocessor technology.
- **Key members:** Samsung, Broadcom, and AllSeen Alliance.
- **Commercial product:** IoTivity

The Open Interconnect Consortium (OIC) is a group of industry leaders who are coming together to deliver a specification, and to promote an open source implementation to improve interoperability between the billions of devices making up the Internet of Things (IoT).

Intel's Open Interconnect Consortium's path to IoT is paved with IoTivity. IoTivity is middleware that sits between the physical radio hardware and the application interface. IoTivity has the objective of relating a device's resources and capabilities to other devices in the application domain with no regard as how any device in the domain physically transports its data. However, this product is still on a very early release to be powerful enough to fulfill many of its goal features. Figure 2.3, obtained from a blog post of the European Research Cluster on the IoT by Fred Eady, exposes IoTivity architecture [27].

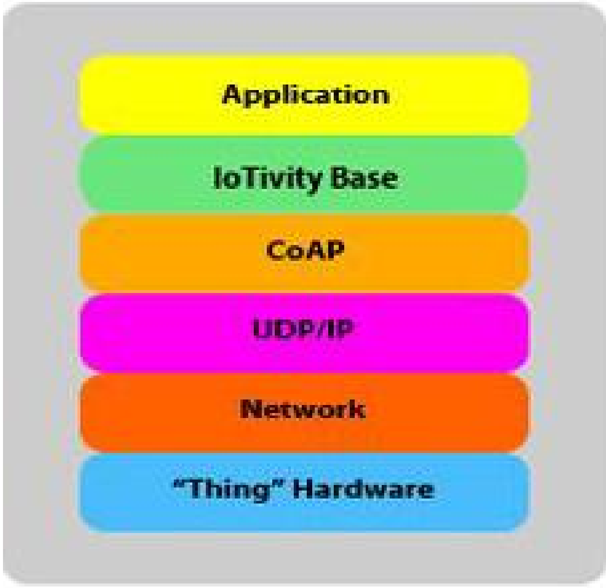

Figure 2.3: IoTivity network architecture

2.3.4 Industrial Internet Consortium

- **Leader:** Intel, the industry's leading supplier of microprocessor technology.
- **Key members:** Intel, IBM, AT&T, General Electric, Cisco
- **Commercial product:** No relevant outcomes announced so far

The Industrial Internet Consortium was founded in March 2014 to bring together the organizations and technologies necessary to accelerate the growth of the Industrial Internet by identifying, assembling and promoting best practices. Membership includes small and large technology innovators, vertical market leaders, researchers, universities and government organizations. IIC attempts the support of a better access to big data with improved of the physical and digital worlds to unlock bussiness value. It is open to any bussiness, organization or entity with an interest in accelerating the Industrial Internet, for now it counts with 50 members.

2.3.5 Thread Group

- **Leader:** Nest (acquired by Google in January 2014).
- **Key members:** ARM, freescale, ATMEL, IBM, AXEDA, XIVELY, Texas Instruments, Toshiba, Sierra Wireless, NEC Engineering, Microsoft, Huawei, Philips
- **Commercial product:** Thread

Thread has a powerful effect over the IoT ecosystem. Partnerships with Mercedes-Benz, Whirlpool Comp and lightbulb maker LIFX to integrate their products with Thread's thermostats and smoke detectors are already in the market. Furthermore, all Nest products already use a version of Thread. After the purchase of the company in 2014 by Google, the former Google division for IoT projects, Android@Home, has been replaced by Thread.

The group has build up a P2P communication system using partially open network standards: IETF 6LowPAN and UDP protocols are applied, but upper layers go on their own way, worked out by the Nest/Google ecosystem, isolating open standards such as CoAP or OMA-LW-M2M. Offering a new way to mesh network, Thread's claim to fame is that it can be retrofitted to the existing IEEE 802.15.4 based devices, supporting their own protocol as the result of improving the existing mesh networking methods, and it is defined as an IPv6 protocol for *smart* household devices to communicate on a network.

Thread network is specifically designed for the home, bounding the target market for applications using their networks, that will be mainly home automation projects. Inside this scope, Thread is designed to support a wide variety of products for the home: appliances, access control, climate control, energy management, lighting, safety and security. It provides AES encryption to close security holes that may exist with other wireless protocols, at network and application layers. Also, it is scalable up to 250 devices onto a single network supporting multiple hops.

Focused on fulfilling all the issues of devices attached to a home automation application, it is aware of battery constrains of those hardware pieces. It offers support for sleepy nodes to offer a long term operation range, even with devices whose drainage is a single AA battery. For the same reason, it keeps the standard IEEE802.15.4 in lower layers (Physical and Medium Access Control) due to its power efficiency. Furthermore, to reduce latency or overhead, *Thread routing protocol* is streamlined. The complete architecture is provided in the Figure 2.4, obtained from the Thread Group website, where technological aspects are exposed [20].

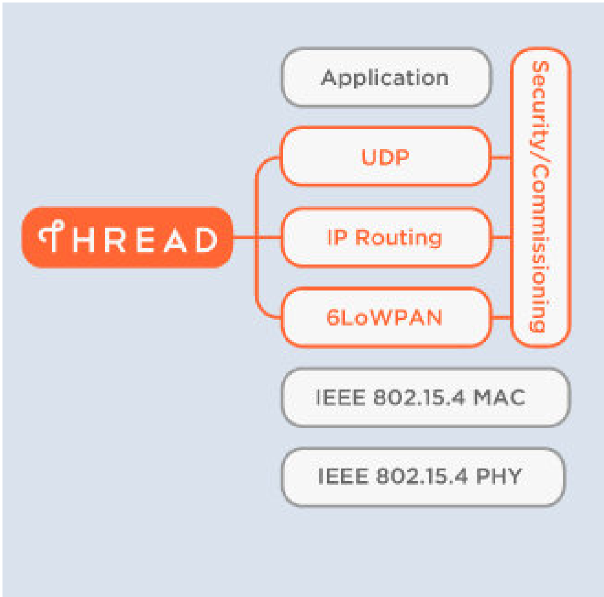

Figure 2.4: Thread network architecture, supporting a house seal protocol for the upper layers

2.3.6 Android Wear

- **Leader:** Google, also the owner of the idea (**Ecosystem owner**).
- **Partners:** Motorola, Samsung, LG, HTC, Asus
- **Commercial product:** Android Wear

A version of Google's Android operating system designed for smartwaches and other wearables. By pairing with mobile phones running Android version 4.3+, Android Wear integrates Google Now functionality and mobile notifications into a smartwatch form factor. It is also linked with Google's Play Store, enabling the possibility to download apps.

It was launched on June 25, 2014, at Google I/O within the products Samsung Gear Live and LG G Watch, along with further details about Android Wear. Later, in September 2014, Motorola released also a product supporting Android Wear: Moto 360.

Android Wear uses Bluetooth Low Energy, so it only works with smartphones running Android 4.3 or newer, less than the 25% of all the Android devices on the market by the time it was released. This faced a handicap for the commercialization of the product, and the competitor Sony took advantage of it throughout the product SmartWatch 2, compatible with older

Android 4.0 devices. Figure 2.5 illustrates the dataflow followed to link devices with Android wearables.

Android Wear includes the Android OS fully displayed inside. Although it brings the power of all the existing libraries, animations and SDK features to a small screen, which is very attractive for potential developers that will find the product as other Android library; inconveniences are hidden in the same idea: a tiny watch supported by a huge OS carries lots of overhead in the transmission, and also battery drainage lack. Figure 2.5 exposes Android Wear dataflow, obtained from a blog post in www.androidauthority.com [28]

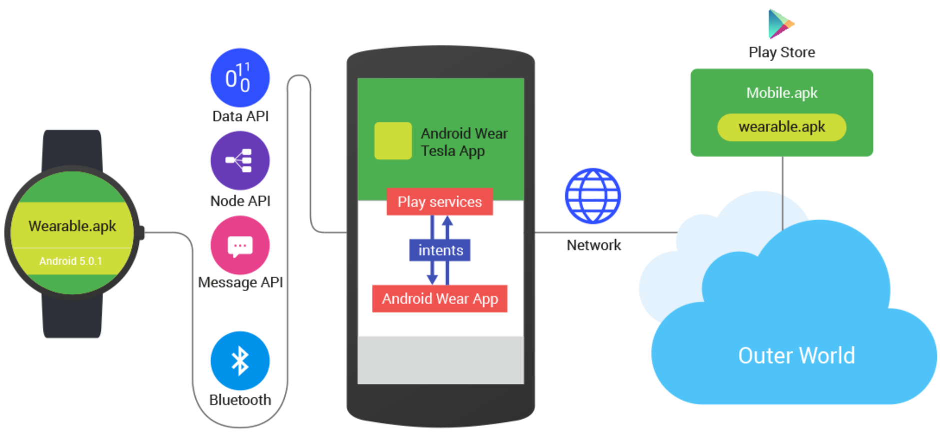

Figure 2.5: Figure illustrating the dataflow diagram of Android Wear.

2.3.7 HomeKit

- **Leader:** Apple, also the owner of the idea (**Ecosystem owner**).
- **Partners:** No partnership acknowledged
- **Commercial product:** HomeKit

Announced in the 2014 Apple Worldwide Developers Conference (WWDC), it is a smart-home platform that enables to control various connected gadgets, group actions and manage audio (AirPlay, CarPlay) from one spot (iBook, iPad, iPhone). HomeKit is a framework for communicating with and controlling connected accessories in a user's home. Users can discover

HomeKit accessories in their home and configure them, creating actions to control those devices. Users can group actions together and trigger them using Siri voice recognition software, fully integrated in HomeKit.

After a partnership announced with the world renowned Mayo Clinic, HomeKit group has launched another product: the HealthKit, which allows apps that provide health and fitness services to share their data with an app developed using HealthKit and with each other. A user's health information is stored in a centralized and secure location, and it is the user who decides which data should be shared with open source apps.

HomeKit is made support iOS 8, and works over Bluetooth Low Energy (BLE, Bluetooth Smart, Bluetooth4.0). It provides seamless integration between accessories that support HomeKit Accessory Protocol and iOS devices, allowing for new advances in home automation. By promoting a common protocol for home automation devices and making a public API available for configuring and communicating with those devices, HomeKit makes possible a marketplace where the app a user controls their home with doesn't have to be created by the vendor who made their home automation accessories, and where home automation accessories from multiple vendors can all be integrated into a single coherent whole without those vendors having to coordinate directly with each other.

HomeKit allows third-party apps to perform three major functions:

1. Discover accessories and add them to a persistent, cross-device home configuration database.

2. Display, edit, and act upon the data in the home configuration database.

3. Communicate with configured accessories and services to get them to perform actions, such as turning on the lights in the living room.

The home configuration database is not only available to third-party apps, it's also available to Siri. This allows users to give commands like, *Siri, turn on the lights in the living room.* If a user creates a home configuration with logical groupings of accessories, services, and commands, Siri can make it very easy to accomplish sophisticated operations with voice control.

In resume, HomeKit is a home automation system very suitable for users with apple devices. Otherwise, it is an iddle system.

2.4 Smart City projects

In March 2015, the famous and influential Forbes group, published an article on their online magazine talking about the top five Smart Cities in the world [32]. The study, based on a list compiled by the consulting company Juniper Research, focuses on two features required on Smart Cities: **sustainability** and **efficiency**. According to those two conditions, five essential components were selected as key comparison term between many existing Smart Cities, those are:

- Technologies
- Buildings
- Utilities
- Transportation and road infrastructures
- The smart city itself

After analyzing each city's *smart* capabilities, focusing on smart grids usage, smart traffic management, and smart street lighting, the leading ones were Barcelona (Spain), New York (US), London (UK), Nice (France), and Singapore. Let's talk briefly about some of them in order to inherit with real examples what a Smart City should be.

2.4.1 Barcelona

BCN Smart City, the name chosen for the project used to transform Barcelona into a Smart City, is an example of success: a pioneer putting into practice the idea of Smart City, Barcelona has an ambitious vision of the city that wants to raise as, and this vision includes a wide range of areas where IoT projects run to fulfill Smart City solutions.

Public and social services

Projects related to the optimization of public and social services, making the city more democratic and promoting a better atmosphere for the citizens living in it. Programmes:

- **Open Government:** Barcelona City Council aims to facilitate the interaction with the government using an electronic administration service, *eAdministration*, where citizens can carry out administrative processes in a quicker and more flexible way. Some of the current procedures available online are self-assessment for tax on vehicles (IVTM) or management of traffic fines (consult them, pay them, or appeal against them).

- **Health and Social Services:** Social care for vulnerable people, with the *Telecare* service, a domestic care service available 24/7. It consists on a device installed at the user's home, connected to the telephone line to a Call Centre. Currently, the capability for this service is up to 70.000 customers.

Environment

Projects related to the application of innovative solutions to environmental management, aimed to make the city more sustainable and efficient.

- **Smart rubbish collection:** The smart treatment and management of waste creates jobs, saves resources and does not harm the environment. Recycling organic waste is essential for achieving this end, and the City Council is working to make all the necessary resources available to the general public in order to promote and encourage this.

- **Smart mobility:** Intended to turn Electric Vehicles into Barcelona's standard mode of public and private transport, 300 free public charge points are distributed across the city. Electric buses and taxis are also available in the city. LIVE (Logistics for Electric Vehicle Implementation) is the platform supporting electric transport in the city. Figure 2.6, obtained from the Barcelona Government website, shows users can see LIVE app [33].

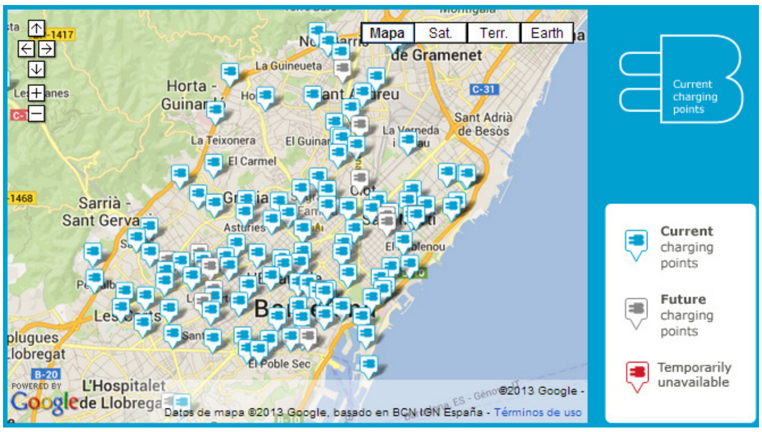

Figure 2.6: Screenshot of the locations of recharging stations in Barcelona, from LIVE project.

- **Smart Water:** Various measures have been set in motion in order to enable the city to intelligently manage its hydrological resources. Currently, ornamental fountains and the water irrigation network of several city parks are monitored to avoid possible incidents, that would be detected fast and could be repaired quickly.

- **Energy self-sufficiency:** The City Council has established an Energy Self-sufficiency Plan, producing its own energy, powered by Endesa.

Companies and business

A smart city is a living thing: experimentation and innovation are essential qualities. Within this creative environment, Barcelona City Council is working to generate employment, promote investment and financing, attract talent and provide help for companies and entrepreneurs.

The program Smart Innovation, encourages transversal projects with a creative and supportive environment, with the intention of carrying out research and keeping up with SMEs and entrepreneurs. Some of these projects are **Barcelona Growth** or **Smart City Campus**, whose purpose is linking businesses, technology and innovation centers, such as universities to co-create and test for urban solutions. Partnerships between companies such as Cisco or Schneider with the Barcelona Institute of Technology are already developing new business ideas.

Research and innovation

The city's system for compiling the information that receives from different sources, and treat it to provide an effective response for the city's services.

- **CityOS:** Figure 2.7, obtained from a paper worked out by the *Institut Municipal d'Informatica* and *Direcció d'Innovació i Arquitectures TIC* [34], exposes Barcelona's techological platform, from the set up to acquire and process information. In another words: the BCN IoT platform. It is a decoupling layer between data sources and Smart City solutions, or front end applications. Consisting in 4 main elements: a main data repository, a semantics layer to unify data access, an ontology (dictionary) to organize data, and a data process container; they are separated on three levels of data: **load** data when it has not been unified yet, a **unified repository** for normalized data, and a **Publication Area** for analized data. All the components within CityOS are open source, so different vendors can plug extra modules to connect extra functions.

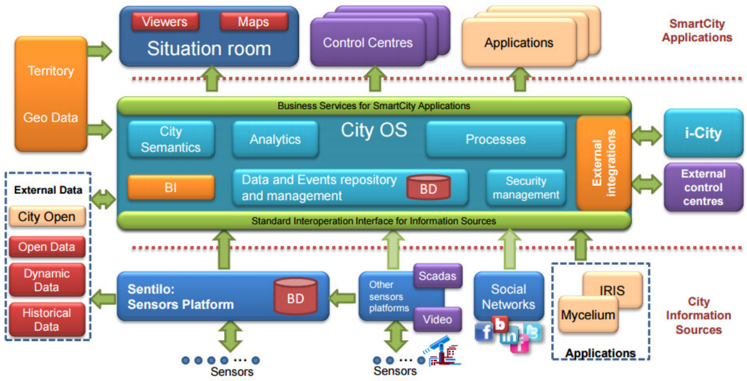

Figure 2.7: Architectural vision of cityOS, the backbone platform supporting BCN Smart City

- **Sentilo:** One of the most powerful ideas powered by BCN Smart city, Sentilo is an open source sensor, an actuator platform designed to fit in the Smart City architecture to exploit the information generated by the city, recorded in the layer of sensors deployed throughout. Considering the amount of devices spreaded through the city, Sentilo includes a non-SQL database to provide scalability. It also supports a graphical interface, provided as public demo, to manage easily the platform. Furthermore, to connect with third party players Sentilo exposes a REST API.

- **Open Data:** The city exposes datasets regarding many of the previous aspects, to be accessible for citizens or possible entrepreneur developers.

Infrastructures

Concerned about respecting the environment, the goal of making the best use of resources and enabling a more sustainable city management through technologies, is also applied in the city's infrastructures.

- **Smart lighting:** This plan prioritises illumination for pedestrian areas, improving lighting and energy-efficiency levels, incorporating technology into the management of public lighting.

- **Smart Urban Mobility:** *Smartquesina* is the project developed for creating interactive bus stops. Smarquesina is touchscreen giving information on bus-arrival waiting times, and with utility apps for enabling users to travel around the city. On top of that, the screen has USB ports integrated into the side, to plug and charge your own devices.

2.4.2 New York

In April 2015, the Mayor of New York City announced the release of *One New York: The Plan for a String and Just City*, a project to develop sustainable and resilient city based in social, economic and environmental challenges. The plan prepares New York City for the future, where digital technology and the exponential growth of data are transforming every urban aspect, the City must respond to these changes and use digital tools to improve services and create more opportunities for its citizens. These efforts are deployed in different areas:

Smart Government and community

Set of projects that put a special effort in improving the service delivery and increase the relationship between administration and residents, enabling a positive feedback loop between them.

- **Snow Plow Tracking:** New York City receives two feet of snow every winter, causing a massive paralyzation of daily routines in the city. To overcome this issue, a tool called **PlowNYC** allows the citizens to monitor snow removal progress in real time. The app implements GIS software, and snow removal tracks integrate GPS flip-phones that are able to send a signal to a data center in less than 12 seconds, permitting a full connection between the service and the app.
- **24/7 Service Requests** The 311 service is a mobile service that facilitates the interaction between the Government and the citizens. Through an app, users send an average of 13.000 messages per day and share frequent answers and questions, creating a good atmosphere where people feels engaged with the city.

Smart Public Health and Safety

Committed to be the safest large city in the US, the New York City Police Department has developed a powerful public safety data collection system, the Domain Awareness System, composed by thousands of NYPD cameras, license plate readers, radiation and chemical sensors, and 911 reports into a telephone switchboard. The goal of DAS is **real time gunshot detection**, deploying a system with hundreds of rooftop mounted sensors, programmed to react against the noise of a gunshot. After identifying the shot, it can be located within a ratio of 25 meters location. Figure 2.8, obtained from the NYC Mayor's Office of Tech and Innovation [38], displays how this app works.

Figure 2.8: Real-time gunshot detection bounds the area where the shot was heard and notifies the 911.

Smart Transport and Mobility

New York's public transportation offers a reliable, safe, sustainable and accessible network that fulfills the needs of all the end users of the public transportation services. Considering the dimensions of the island, one of the goals of Smart transportation is reaching every single point within the city in 45 minutes or less.

- **Responsive Traffic Management:** Midtown in Motion is a technology enabled traffic management system that uses real traffic information from different sources to monitor and respond to various traffic conditions. All the information monitored is passed to the Traffic Management Center, to quickly identify congestion issues and adjust signal timing. It also helps the environment trying to avoid vehicle idle times (times when vehicles are stuck in traffic), which causes a big amount of GHG and pollution.

- **Traffic Signal Priority:** Equipping buses and traffic lights with real-time sensors that prioritize bus transit through signalized intersections, bus delays have been reduced in a 20%, making public transportation more attractive for citizens. The mechanism works with a GPS and location -based traffic control software built into the buses and into the traffic controllers, communicating with each other and also with the Traffic Management Center. Figure 2.9 explains the behaviour of this disruptive idea. Figure obtained from the NYC Mayor's Office of Tech and Innovation [38].

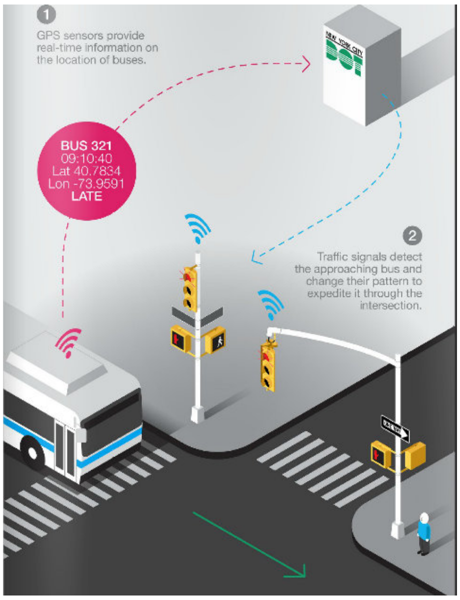

Figure 2.9: Traffic Management for public buses in the Smart New York City

Smart Energy and Environment

New York city's water supplies come from 19 different reservoirs and three controlled lakes, which combined store a capacity of 580 billion gallons of water that drain almost 7000 miles of pipes and aqueducts. All this water is monitored by the Department of Environmental Protection (DEP) through an extensive network of remote monitoring sensors, providing 24/7 information about the city's water supply, alerting about possible quality issues or other anomalies.

For private infrastructures, the DEP has installed an Automated Meter Reading (AMR) system, that collects measures and sends the information to a low-power radio transmitter, connected to the Network Operations Center. Data is analyzed there, and customers receive real time billing information. They can even pay water bills online, and more important, they are notified when anomalies such as leaks occur.

Chapter 3

Platform architecture and description

Within the scope of European Commission targets for H2020, an open initiative is released under the name of **FIWARE** project, aiming to create a sustainable ecosystem after the integration of many Internet technologies. FIWARE introduces itself as an open alternative to existing proprietary Internet platforms, following the *Open APIs for open minds* core value. The initiative bases itself in five pillars including the physical implementation of an IoT platform accessible through APIs, or an accelerator program for possible stakeholders. Linked to this program, the EU launched a campaign in 2014 in order to approach entrepreneurs or small companies to develop innovative applications based on FIWARE. After the success, the program is meant to be extended worldwide in the future.

FIWARE

The FIWARE platform provides more than a powerful set of public APIs: it also eases the development of Smart Applications in multiple sectors, and it offers as well an open source reference implementation of each of the FIWARE components, for potential providers to emerge faster in the market.

FIWARE Lab

FIWARE Lab is the non commercial sandbox environment where experimentation and testing based on FIWARE technologies take place. It is the background layer of the whole development, where entrepreneurs and developers can implement their applications.

FIWARE Ops

Collection of tools that enables the deployment, setup and operability of FIWARE instances by Platform Providers, that also allow to help expanding the infrastructure to additional nodes (datacenters). FIWARE Ops is the tool used to build, operate and expand FIWARE Lab.

FIWARE Accelerate

With economic purposes rather than technical, FIWARE Acceleration Programme encourages possible developers, focusing on Small and Medium Enterprises (SME) or Start-ups, to adopt FIWARE solutions as the solution for their project. This program is founded by the EU with a large sum of money.

FIWARE Mundus

Although spreading the FIWARE product is yet an ambition, FIWARE Mundus is designed to bring coverage to different parts of the world, including Latin America, Africa and Asia.

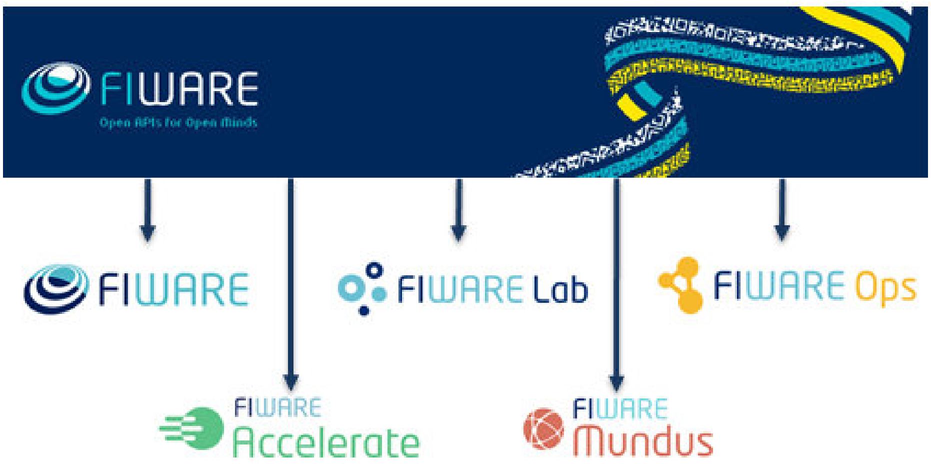

Figure 3.1: FIWARE initiative and its five pillars

Figure 3.1 includes an scheme of how FIWARE initiative is segmented. Icons are corporative and obtained from the official FIWARE website [39].

FIWARE is implemented through the provision of enhanced OpenStack-based cloud hosting capabilities, plus a rich library of components (called, from now on, "Generic Enablers") accessible through open standard APIs, that make easier to connect to IoT devices, process data and media in real time at large scale, and also perform Big Data analysis or incorporate advanced and very visual features to interact with the user. Generic enablers can be found also mashed up in bundles, where a set of generic enablers that perform common tasks can be deployed at the same time.

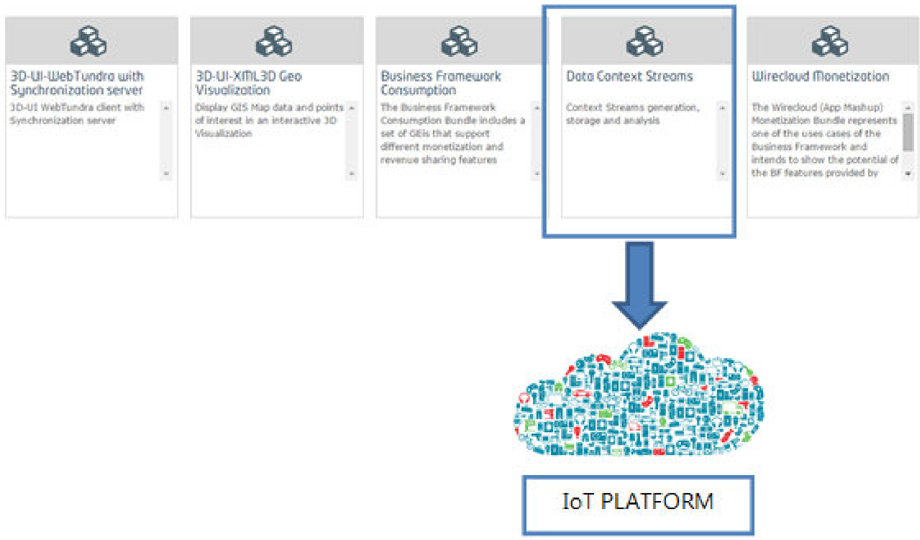

Figure 3.2: Bundles in the FIWARE Platform

In order to build an IoT platform suitable to process data, store and analyze it, the appropriate FIWARE bundle to deploy is the **Data Context Streams** bundle, as it is explained in Figure 3.2, where the figure representing several data bundles has been taken from the official FIWARE website [39]. Data Context Streams bundle is the base to build a software architecture suitable with the availability and accessibility that any IoT solution needs to fulfill. Through the exploited components introduced in the following block diagram, it will be possible to gather information at large scale, produce, publish or consume it to transform the information into a more reliable one, bringing to light some non-implicit contexts from the former ones.

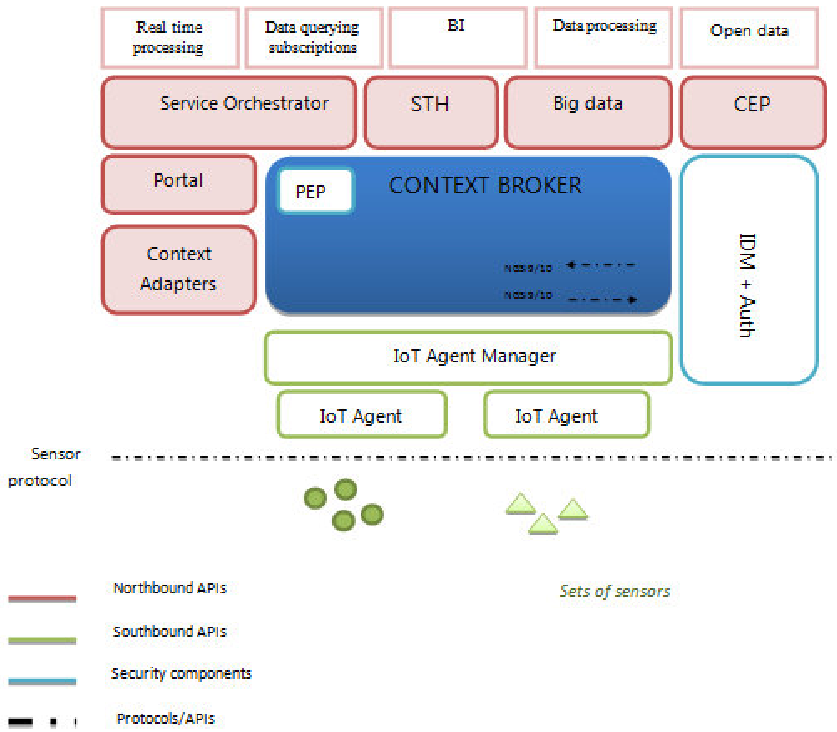

Figure 3.3: IoT Platform based in FIWARE bundle

This architecture represents the whole IoT platform, where a central component, *Context Broker*, manages incoming and outgoing information through NGSI APIs. All the components in red are part of the *Northbound platform*, and they are attacked through the same API call, while the components in green, representing the *Southbound platform*, use a different one. The *Southbound platform* is the grid that manages all kind of interaction with sensors through a gateway that understands the native sensor protocol and transforms it into a NGSI call. Finally, the components in blue represent the security parts of the platform: on the one hand, a secured access to the Context Broker itself; on the other hand, a security scheme based in auth token and required for all the calls to all of the components. Orchestrating the whole system is done through a simple **multitenant** model, where a **FIWARE service** and **FIWARE service path** are the tenants to access to a private context shared among all the components. Of course, all the sets of services and service paths are idle, so any operation of any of the components can listen to the other sets. Since the whole platform makes its calls through API rest, tenancy information is transmitted in the shape of HTTP headers.

In the following subchapters, a wider vision for each of the components available in the platform will be introduced.

3.1 Orion Context Broker

The **Orion Context Broker GE** is able to handle context information at large scale by implementing standard REST APIs. This is important because in many of the cases, the income context information will arrive from many different sources, and it needs to be gathered anyway: from sensor network grids, to users through their mobile apps, or databases storing big pieces of information. Thus, one of the most important features of Context Broker is that the access to the information independently from the source of information. So, if the source suddenly changes, the information can still be retrieved. In summary, Context Broker allows the following operations:

- **Register context** producer applications, e.g. a temperature sensor within a room.
- **Update context** information, e.g. send updates of temperature.
- Being notified when changes on context information take place (e.g. the temperature has changed) or with a given frequency (e.g. get the temperature each minute).
- **Query context** information. The *Orion Context Broker* stores context information updated from applications, so queries are resolved based on that information.

To implement the operations specified above, *Context Broker* is able to transmit or receive data depending on which role it is working as: on the one hand Context Broker would act as Context Producer, storing new information or updating the one available: using another words, producing information and gathering it. On the other hand Context Broker would act as Context Consumer, exporting the available information somewhere else, through a valid subscription. Both roles are handled by API REST with the interfaces *NGSI10* and *NGSI9*.

Context Broker implements a mechanism so part of the context information is not managed directly by itself, but by an external source, that would be the Context Provider. In this case, *Context Broker* will act both as Context Consumer and Context Producer: Context Consumer of the incoming information from the external source, and Context Producer to export the information somewhere else. The advantages of this feature are the easy interaction for client applications just through one API endpoint instead of using a different one for each source of context, the scalability given to the component when it is not able to manage certain information by itself and it is easier to access it from somewhere else, and the security that gives to the whole platform the fact that clients may not be allowed to access directly to external Context Providers due to security reasons. The scheme below represents this mechanism.

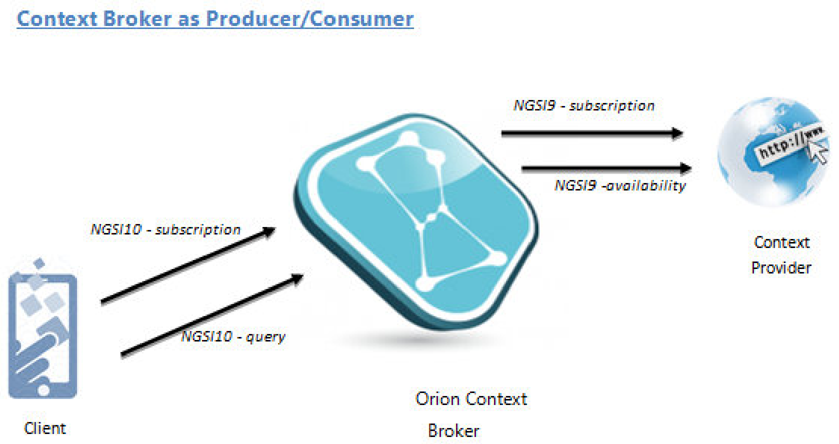

Figure 3.4: Context Broker role as context Consumer and Producer at the same time

Playing a double role, *Context Broker* acts as a consumer of data, subscribing itself to the Context Provider, which would be a third party site or app. The subscription could either be a subscription on time, to be updated after a convenient time; or a subscription on change, where the entity is updated as soon as the data changes in the source. In order to publish the subscribed information in *Context Broker*, update operation is performed according to the subscription details predefined, towards the desired context entity. *Context Broker* also acts as a provider of data when the role of Context Producer is adopted, if a client requests context information published in *Context Broker*. In this case, the client subscribes to *Context Broker* (now as context Provider), who sends the information that was previously received by an external site, and published as context data. The result of the whole operation is the creation of a link between the external site (former provider) and the client interface, established via Orion Context Broker.

3.1.1 NGSI standard operations

A list of basic operations and a short description is available here:

NGSI9: Manages registrations and discover context availability	
registerContext	Registers entities coming from external sources
discoverContextAvailability	Accesses information about the availability of the context source where the entity information is stored
subscribeContextAvailability	Aynchronous notification about the availability of the context source
unsubscribeContextAvailability	Opposite operation to the previous one. If there is not subscription operation, context availability must be discovered using polling
NGSI10: Manages context consumer and producer features	
updateContext	Creates an entity with some information (APPEND), or updates the information available of an existing entity (UPDATE)
queryContext	Access the context information of an existing entity.
subscribeContext	Asynchronous subscription to update or append operations, based on time or on change. As result, an updateContext operation will be launched when the subscription conditions are fulfilled
unsubscribeContext	Opposite operation of the previous one. If there is not a subscription operation, information will be updated or appended using polling

3.1.2 Notation in Context Broker

Context Broker establishes a specific terminology to refer all the data inside of its context. For instance, the virtual representation of a thing in real life, would be an entity inside Context Broker, and the different characteristics of this thing, would be attributes. Every entity or attribute carries an id, a name and a type to reference it inside of the whole context. Furthermore, in relation with the multitenancy feature provided by FIWARE, every set of entities entity (and their attributes) would be encapsulated inside of a Service and Service Path, to whoms the entity would belong.

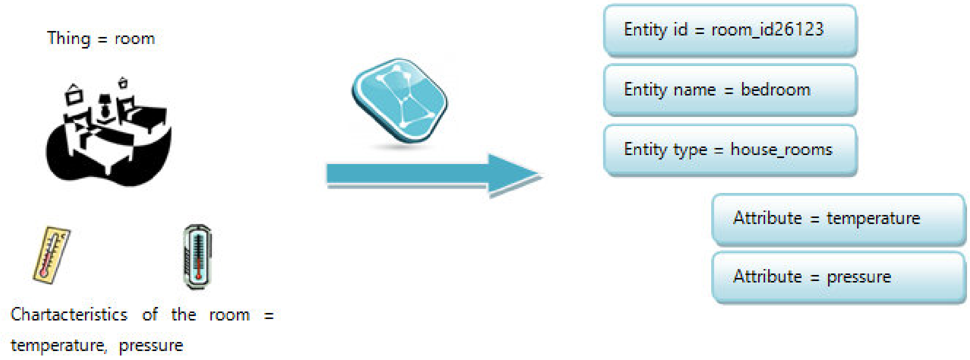

Figure 3.5: A physical device is manipulated to acquire the shape of a context entity

JSON payloads

JSON, JavaScript Object Notation, is a lightweight data-interchange format, very easy to read and write for humans, and also easy for machines to parse and generate. It is defined as a text format completely independent from programming languages, but with common properties with many of them. For that reason, the previous entity should be understood as a JSON structure as well, and it would have the following shape:

```
{
    "type": "house_rooms",
    "isPattern": "false",
    "id": "bedroom",
    "attributes": [
    {
        "name": "temperature",
        "type": "float",
        "value": "25"
    },
    {
        "name": "pressure",
        "type": "integer"
        "value": "720"
    }
    ]
}
```

Geolocation attributes

Entities can have a location, specified by one of its attributes. In order to state which attribute defines the location, a *location metadata* is used. Metadata information specifies a sub-condition fulfilled by the attribute, which in the case of the location, is the standard World Geodetic System (WGS 84). Here is an example of how a location attribute is parsed:

```
{
    "name": "position",
    "type": "coords",
    "value": "40.418885, −3.691944",
    "metadatas":[
    {
        "name": "location",
        "type": "string",
        "value": WGS84"
    }
    ]
}
```

To avoid inconsistencies, *Context Broker* only accepts one attribute defined as location per entity. The value of the location is a string with two numbers separated by a comma, denoting latitude and longitude respectively. Notice that degree-minute-second notation is not allowed.

3.1.3 Database Storage

MongoDB is an open source document database that provides high performance, supporting embedded data models that reduce the I/O activity; high availability, through replica sets (servers that maintain the same datasets, providing redundancy and increasing data availability); and automatic scaling, with horizontal scalability as part of its core functionality. MongoDB is document-oriented, and every record (file) inside is represented in a format very similar to JSON *Orion Context Broker* publishes context information that will be stored into a MongoDB database, that must be running while any process is open in Context Broker. Due to the geolocation capabilities of Orion Context Broker, the version of MongoDB required is 2.4 or higher in order to use the geolocation features available.

3.2 Policy Enforcement Point

To secure a backend service, *Context Broker* includes a Policy Enforcement Point (*PEP*) proxy, that allows the validation of the requests the before sending them to the service. The validation will be done against the *Access Control* component (the other security component available in the platform), and it is based in several pieces of data:

- **User Token:** Token coming from OAuth2 authorization server at the *Access Control*. Carried in the X-Auth-Token header.
- **Service ID:** Read from the FIWARE service header, identifies the protected component.
- **Subservice ID:** Read from the FIWARE service path header, identifies further divisions of the previous service.
- **Action:** The action after a particular request is guessed by checking the path or inspecting the body. If the logic of the response depends on a secured componets, an special access plugin will be required to access to the component in order to perform the action.

The authentication process is based on OAuth v2 tokens, and *PEP* Proxy currently supports two possible authentication authorities: Openstack Keystone or Keyrock IDM. While Openstack Keystone is an open source software boosted by openstack.org; Keyrock IDM is the FIWARE house seal, fully compatible with the previous for easy use in applications using other FIWARE components.

3.3 Access Control

Access Control tool enables the definition of several roles that would determine which types of users can access which specific resources in a backend. Roles may have different role hierarchies or specific permissions for each, and even more complex role permissions such as combining a certain role with other user's role attributes. In any case, the tool to assign a role to a user is the Identity Manager Interface (IDM). The Identity Management tool covers a number of aspects envolving access to networks, services and applications, including secure and private authentication from users to access those networks, services (Single Sign-On) and applications (Identity Federation)

3.3.1 Single Sign On

Single Sign On is suitable for *Access Control* over independent software systems that are related. Using SSO, the user would log in with a single ID to gain access to the whole pile of systems without being prompted for different passwords. Typically, it is performed through LDAP protocol, an open standard protocol in the application layer of the Internet protocol suite, for accessing and maintaining distributed directory information services over an IP network

3.3.2 Identity Federation

Identity Federation allows SSO without passwords: the federation server knows the username of an user in each application, and shows this information with a token validating it. This token is OAuth token, and due to the trust between the two interconnected systems, the target application would accept the token and authenticate the user.

3.3.3 Security in the platform

Figure 3.6 offers a wide vision of the security in the platform is graphically explained, obtained from FIWARE wiki [40]:

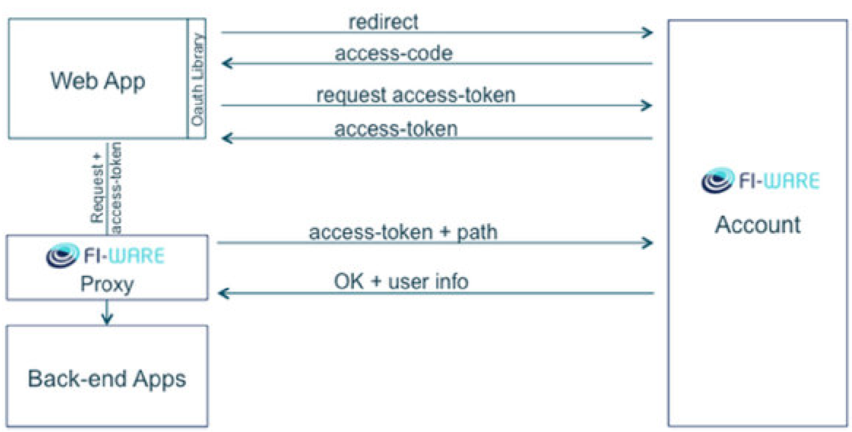

Figure 3.6: FIWARE account interacting, on the one hand with PEP and PEP proxy; on the other hand, with Access Control via OAuth

Within the context of an *Access Control* environment, the FIWARE account (i.e. the FIWARE components of the IoT platform) would interact with an external web app through an OAuth identity validation, where the steps followed in order to communicate both systems. First of all, data from the third party site redirected to the platform, that would request an access code to the external site. The site would query then a token as a response of the previous request, which carries the role permission details. When the platform sends the token, communication is stablished if the permission is valid.

Furthermore, a system of validation via auth token plus tenant information (path) is requested to validate the interaction of components inside the platform, via PEP proxy.

3.4 Context Event Processing

Starting to describe the northbound components, the first functionality to be taken into account when developing an IoT application is a real time processing of data, and the *Context Event Processing* module is suitable for this operation. *CEP* enables the detection of patterns above contexts, and establish a reaction to them. As input, it receives context information as events, and generates observations or situations as output events, after analysing the input event data in real time, generating inmediate insight and enabling instant response to changing conditions.

Applications connected to the *CEP* can play two roles: the role of Event Producer or the role of Event consumer, and both roles can be performed at the same time for an application. When the application is an Event Producer, events can be provided by means of invoking a REST API call (textbfpush mode) or they can be provided through a REST API that the *CEP* can invoke to retrieve events (textbfpull mode). Event Consumers are the destination points of events, and the typical endpoint for events in this IoT platform would be the Context Broker connected to the *CEP* as an Event Consumer, forwarding the consumed events to all the interested applications based on a subscription model.

After an event is received in *CEP*, a pattern is selected from the event with a time window or segmentation methods. Predefined patterns can be built using a Web based tool (Portal component, it will be explained later in this chapter). Patterns can go from status flags following a sensor context (failed, fixed, succeed), to alarms if within a time window no sensor events are arriving meaning that the source is down. Every pattern is associated with an Event processing context, that groups event instances so they can be processed in a related way. The Event processing context can be a temporal processing context if the pattern of an event is a time window, or a segmentation processing context if the pattern of an event is an attribute (as an attribute inside an entity of Context Broker).

It is quite clear how *CEP* component and *Context Broker* are correlated. In fact, the major interaction between them consists on a loop where *Context Broker* acts as a Event Producer and Consumer towards *CEP*. The scheme below illustrates this complex scenario.

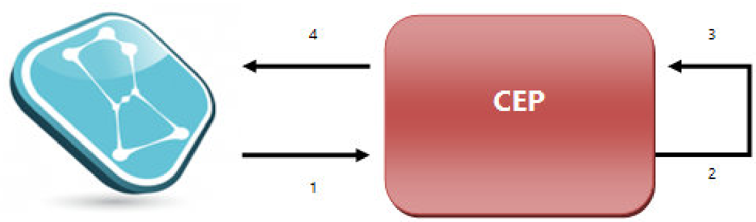

Figure 3.7: Context Broker can interact with CEP as Context Producer and Context Consumer at the same time

1. Data output from *Context Broker* is launched as an event to the CEP: *Context Broker* is an Event Producer, and also a Context Producer.

2. After seeking for the trigger pattern, data output may receive, as part of the event reaction, the order of invoking back the transformed event into another *CEP* operation: in this case, it acts itself as an Event Provider.

3. Once the output event has been received as new input, a different pattern will obtain insight information, which will be sent to *Context Broker*.

4. *Context Broker*, now receiving an event as data, is an Event Consumer, and also a Context Consumer

As an example of the described scenario, let's imagine a sensor dataset as input for *CEP* in [1], where a pattern would be based on a time window measuring the delay between two consecutive events. Between flows [2] and [3], the delay information would be interpreted as an attribute, and filtered with the second pattern to reject events with high delay, that would not go back to Context Broker in [4]. With this brief example, it is noticed how *CEP* is not only an Event Manager: the fact of applying patterns carefully, and designing good links around, transform this component into a powerful insight discoverer tool.

3.5 Short Term Historic

Inside an IoT application, processing information in real time is very frequent, whereas there might be cases where recording past information can be very useful. For those situations, the Short Term Historic tool is intended to provide aggregated time series information about the evolution in time of entity attribute values registered in Orion Context Broker. The STH can directly subscribe to the *Context Broker* to receive notifications when the entity attribute values change (from *Context Broker* operation subscribeContext as Context Provider), calculating the aggregated time series information and storing it.

- **Range:** The scale of time of the aggregated information: *year, month, day, hour, minute.*
- **Aggregation period:** Conceived as the resolution of the time period, the ranges can be paired to offer a more detailed time scale. (*year + month, hour + minute, etc.*)
- **Origin:** Timestamp containing the origin of time for which the aggregated time series is applied. The format is the UTC format (ISO 8601), as in *2010-01-01T00:00:000Z*
- **Offset:** For an aggregated time series pair, difference between them in the range scale chosen.
- **Samples:** Number of time series data available, fulfilling a certain range, aggregation period, origin, and offset.

3.6 Big Data

3.6.1 Apache Hadoop

Before talking about the Big Data tool available in the FIWARE IoT platform, it is important to highlight several concepts related to open source Big Data tools. Apache Hadoop is an open source software framework for distributed storage and processing of very large data sets, based on a modular cluster distribution with several nodes. The core of this software consists on a storage part, HDFS (Hadoop Distributed File System); and a processing part (MapReduce). Hadoop splits the stored files into large data blocks, and spreads these blocks into different nodes, that will be processed in parallel when processing the whole data, making a faster and more efficient approach than conventional data manipulation. HDFS storage system is distributed and scalable, and stores large files (in the range of gigabytes or terabytes) across multiple machine. Although redundancy is not recommended when a system should be processed fast, HDFS achieves reliability as a storage system by replicating data across multiple hosts. Also, for security reasons, the HDFS includes a secondary namenode, that connects with the primary one bultiding snapshots of it for saving into local or remote directories, and in case of failure in the primary namenode, data can be recovered from those images.

3.6.2 Apache Hive

Hive is a data warehouse infrastructure built on top of Hadoop for provigin data summarization, query and analysis. For making simple queries, a language familiarized with the standard SQL language is provided: HiveQL, that reads and transparently converts the queries into MapReduce procedures.

3.6.3 Big Data tool in the IoT Platform

Context information published in *Context Broker* can be stored into an HDFS based database in order to create a long term historic database that enables a wider vision of the context information insights. In the IoT platform environment, a typical example to explain the usage of this tool, is an scenario where massive information, for instance, the one gathered from sensors spreaded in a city over a long period of time, has to be analysed. In order to analyse the data, the first step is uploading it into a web format of HDFS (WebHDFS), where data is stored in a CSV-like format, and queried after through HiveQL calls.

3.7 Context Adapters

A typical problem when interacting with external sites or third party applications is adapting to the data format that they listen. Usually, third parties can be only fed by a certain data type not suitable with the context information published in *Context Broker*. For bringing down this barrier, *Context Adapters* work as connectors between context information and the desired output format. In order to link context information with a third party, the most convenient way is subscribing the application in *Context Broker* through a regular subscribeContext operation (previously explained in this chapter). Once the subscription is made, whenever any data meets the subscription conditions (On Time interval, or On Change), the *Context Adapter* is authomatically fed by the context information, storing and transforming the incoming parameters into the desired third party data format. Usually, third party applications or external web sites using REST APIs include information for developers, offering credentials, API endpoints, and guides with accepted queries towards their software.

3.8 IoT Agents

Real applications connected to the Internet of Things have the need to overcome a set of problems arising in the different layers of the communication model. Due to the lack of globally accepted standards, the interaction with physical hardware pieces faces a heterogeneous environment where families of devices are dispersed and accessible through multiple wireless technologies. Furthermore, devices communicate through different protocols, too specific and using different data encoding languages, which makes impossible any attempt of integration. On top of that, devices are resource constrained and they cannot use full standard protocol stacks: they might not be always available (an essencial feature when talking about an IoT environment) and they cannot transmit information too frequently, all in terms of saving battery. As providing a solution where one size fit all is not feasible, it is tricky to find a global deployment, as a southbound integrator component. Instead, a modular system with several agents listening to different protocols is designed. Each *IoT Agent* deals with security aspects, and connects the native language spoken by the device with *Context Broker*, to whom the *IoT Agent* is able to talk either as Context Consumer or Context Provider. On top of the modular structure, a guided module as a common framework, leads advanced tasks as monitoring or managing the IoT Agents. This common piece, named *IoT Agent Manager*, establishes the classes of devices that the *IoT Agent* can publish, and launches operations towards *Context Broker* once devices have been turned into Context Broker entities inside the *IoT Agent*. Regarding the behavior of a device in nature, three types of parameters are distinguished inside a device:

- **Active parameters:** Data sent from the device is authomatically pushed, and listened by the *IoT Agent*.
- **Lazy parameters:** Due to battery drainage, data is only sent after requesting it.
- **Actuator parameters:** Devices contain an upper grade of intelligence, able to change their status or a certain feature after being told so. For instance, traffic lights swapping from red to green

3.8.1 IoT Agent Manager

It is responsibility of the *IoT Agent Manager* to provide the proper communications with *Context Broker* for these devices to be listened, and published as context. Once again, depending on the sort of incoming information, *Context Broker* may be context Producer or Consumer, but this time, the *IoT Agent Manager* is the unique data source or data endpoint interacting.

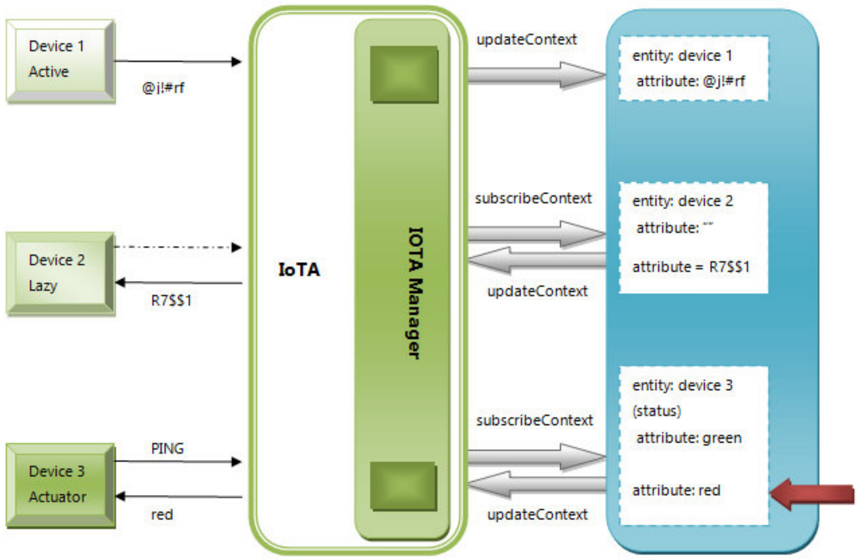

Figure 3.8: Flows linking devices with *Context Broker*, all possible cases

The scheme represents the communications of the three different types of parameters (one per device in this case, although one device can carry several types) with the *Context Broker* via *IoT Agent Manager*. It is not necessary to publish an entity in *Context Broker* before Device 1 talks, whereas a previous subscription is required for Device 2 and Device 3.

3.8.2 IoT Agent Protocols

At present, FIWARE IoT platform supports the following protocols, for which *IoT Agents* are designed:

- **UL2.0:** Protocol over HTTP as a simplification of the former protocol Sensor ML (SML). Ultralight 2.0, shortly named UL2.0, is an extremely simplified protocol identifying observations with alias and therefore reducing the transmission overhead from devices to the minimum.

- **MQTT:** Formerly named MQ Telemetry Transport, this protocol supports transmission over TCP as one of the options suggested by OneM2M. MQTT handles bidirectional connections by establishing a TCP permanent socket to be maintained with keepalives. Observations (measures) and commands (actuator measures) are organized in topics.

- **LWM2M:** Supports OMA LightweightM2M over IETF CoAP for machine to machine communications or IoT device management. LWM2M enabler defines the application layer communication protocol (IPv4 or IPv6) between a server and a client, which is located in a LWM2M device. Target applications of this protocol are devices with a resource constrain, such as limited battery supplies.

3.8.3 Database storage

After a device is mapped in a *Context Broker* entity, a storage system inside the *IoT Agent Manager* preserves this information for further communications with the same device. Persistent in memory, non relational, and document-oriented in favor of JSON-like files, Mongo DB is the storage system chosen to deposit information regarding *IoT Agents*. Files containing information about the entity matching of a certain device, in JSON shape, are stored into a database named as the tenant information linked to that device/entity. So, when a device that has been already stored into the database aims to communicate with the *IoT Agent*, the field in which its information lays, is updated with the new data arrived.

3.9 Orchestrator

As its own name implies, *Orchestrator* component tries to group all provision operations for an IoT platform that typically implies several steps or several systems interaction. *Orchestrator* exposes a single API and provices commands to perform all the operations related to security (multitenancy queries, role assignment); *Context Broker* (NGSI9, NGSI10 standard operations); or *IoT Agent Manager* standard provision queries.

3.10 Portal

Offering a wider vision of the platform, a web portal tries to perform a similar feature as the *Orchestrator* component, but based on data management through data visualization. *Portal* contains tenancy management, where security roles can be defined for different multitenant levels. Furthermore, inside of each FIWARE Service and FIWARE Service Path, *Portal* offers a device manager, an entity manager and *CEP* rules management tool. Inside this project, a detailed resume of portal functionalities can be found in Appendix A.

Chapter 4

Design and development of a Smart City based on the FIWARE IoT Platform

There are many reasons why FIWARE IoT Platform was selected amongst other platforms for this project. Maybe, the most important one is that FIWARE offers an open platform alternative that enables the creation of an open ecosystem around it, permitting the connection with other open tools, from open-source relational database management systems (RDBMS) such as MySQL, to software designed to store and process very large data sets in clusters such as Hadoop (indeed, the Big Data tool of the platform is a component based in HDFS storage system designed by Apache). But this is just an example, because the existence of an open platform alternative will ensure that application providers will be able to choose who will provide and operate the environment where their applications will be hosted. Data providers, including Open Data providers, will also be able to choose who will provide and operate the environment where their data will be hosted and exploited. Their decisions can be driven not just based in economic savings but the trustworthiness of the platform provider. Applications and Data providers can also protect their investment better because of the ability of porting applications and data to an alternative platform provider if a given platform provider stops meeting their requirements, avoiding getting locked in a given platform provider.

The platform is also adaptable enough to host data from many data sources. In fact, several **Service Hosting Providers** provide and aoperate the hosting infrastructure on top of which services are deployed, offering Cloud Services for hosting applications and enablers. In fact, this project is supported inside a cloud server in Amazon Web Services and later in this chapter this will introduced an explained.

Another of the facilities powered by FIWARE IoT platform is adaptability to protocol standards. Middleware components support several protocols on each of the transmission layers; for instance, transport layer supports TCP and UDP, and wire protocols supported can be HTTP or RTPS. Furthermore, IoT Agents working on the southbound platform can listen, among others, to Lightweight M2M that suports OMA over IETF CoAP in the application layer. In the Internet layer, IPv4 and IPv6 are avalable. But all these choices are deployed in a modular architecture, so the platform is scalable by design, depending on how many resources are required to expose.

Multitenancy is an advantage as well, specially thinking on the mesh of data published in an environment as big as a city. Structuring areas and datasets as services and sub-services, each group will be idle from the rest of them, so data will remain private from undesired listeners. Also, the management of Context within the platform has a well organized ontology, very flexible and verbose.

To sum up, due to the previous conditions, FIWARE is suitable when the scenario is big and complex, like a Smart City or an Industrial IoT scenario. For smaller environments, specially the typical case of bringing IoT to the home, other platforms adapt better to a solution.

4.1 Development of the project

Datasource of the project

In the past few years, many cities in the world have exposed their data for the for the benefits of their citizens, SMEs, start-ups and entrepreneurs, or single developers who may find it useful. This project has taken this as an advantage, and the city of San Francisco has been chosen as the scenario where data will be handled and published in the platform, obtaining data from an open data portal developed by San Francisco Government, launched in 2009, and containing more than a hundred datasets for open and public usage.

The reason why San Francisco was chosen as the real environment where this project is settled, is the city's leadership in Smart city ideas, that takes it to publish many relevant data useful for a project like this one. Also, two extra factors were essencial to select the City by the Bay: **the awareness of the city and citizens to preserve San Francisco as a green and clean spot, with MUNI, San Francisco Municipal Railway, as a great example of public transportation in a US city**, and **the particular**

conditions of the region, a land conditioned by many environmental accidents that have affected the atmosphere and history of the city, forcing it to be a step ahead in controlling possible natural disasters.

A model of Smart City

After reviewing the concept of Smart City, also looking the current trends followed by pioneer real models of Smart Cities such as Barcelona, London or New York, and attending to the standardization requirements to fulfill a Smart City model, the following summary of the services contained in the urban design can be supported by five pillars, as shown in Figure4.3, obtained from the website of the magazine *ecointeligencia* [52].

- Smart Mobility
- Security and eHealth
- Energy and Environment
- City Economy
- City Management and eGovernment

Figure 4.1: The five essential components that conform a robust Smart City, and some examples of applications attached to each of them

Of course, with the available data, it is not possible to fulfill a model as robust and complete as the proposed within the scope of this project. However, datasets were carefully chosen trying to expose at least one scenario for each of the features that would compose the platform.

Smart Mobility: Bicycle Parking	
Bicycle parking installed throughout San Francisco that is free and available for public use.	Format JSON
Security and eHealth: Health Flu Shot Locations	
List of San Francisco Department of Public Health clinics offering flu vaccinations throughout the city.	Format JSON
Energy and Environment: Traffic Noise	
Modeled average daytime and nighttime noise levels from traffic (in Decibels) in San Francisco.	Format SHAPEFILE
City Economy + City Management: Restaurant Scores	
The Health Department has developed an inspection report and scoring system of restaurants and venues in the city.	Format CSV

Table 4.1: Datasets obtained as resources in the different areas of the Smart City for the scope of this project

These datasets will be the information reference taken for the scope of this project, that will simulate the sensors from which all this data is stored in real life, in order to publish and handle it through the FIWARE IoT platform. In the following parts of this chapter, the process to convert these data into real data collected from sensors will be explained, to later send data as real data from a sensor using a M2M communication protocol, and finally publish and show the captured data using a data visualization tool linked as Context Adapter within the application environment.

4.1.1 Transforming data into real sensors

Bicycle Parking

Bicycle Parking data set carries information regarding to each spot to park them, including the number of parking slots and racks, the placement where the bike is stored (GA = garage, GAC = garage cage, LOT = within a property, RD = roadway, SW = sidewalk), the location name, coordinates and date when it was stored.

```
{
    'spaces': '38',
    'placement': 'RD',
    'street_name': 'GROVE',
    'yr_inst': '24GROVEST',
    'addr_num': 'SFMainLibrary',
    'mo_installed': 'UK',
    'latitude': {
        'latitude': '37.77915913',
        'needs_recoding': False,
        'human_address': '{
            "address": "GROVE",
            "city": "",
            "state": "",
            "zip": ""
        }',
        'longitude': '-122.41580788'
    },
    'racks': '19',
    'yr_installed': '2005'
}
```


But some of this information is not relevant when the aim is sensing where bikes are at each time in the day. Then, a random model based on the on/off peak times times in the day, considering different slots associated to a greater probability of movement per bike has been established to generate a set of bicycles travelling through the city, they will be provided with a simulated location based on time, storing a measure every half an hour.

BICYCLES DURING A DAY	
Morning peak: 6.30 to 9.00	An average of 75% of bikes on movement
Morning regular: 9.00 to 12.00	An average of 35% of bikes on movement
Noon: 12.00 to 15.30	An average of 55% of bikes on movement
Evening peak: 15.30 to 19.00	An average of 70% of bikes on movement
Evening regular: 19.00 to 21.30	An average of 40% of bikes on movement
Night time: 21.30 to 6.30	An average of 10% of bikes on movement

Table 4.2: Data model applied to the bicycles travelling along the city

Taking the information in JSON from the bicycles, 40 bikes will be simulated through the city according to the averaged percentages bikes on movement per hour, and the information about *sensored* bikes will be stored into a csv file.

19/08/2015	0:00:00	0.30:00	1:00:00	1:30:00	2:00:00
bikeasset002	37.752885/-122.411477	37.764391/-122.433406	37.768682/-122.420335	37.790937/-122.401337	37.765998/-122.449678
bikeasset003	37.752455/-122.412221	37.76962/-122.422059	37.763513/-122.483856	37.74174378/-122.406661	37.756417/-122.421372
bikeasset004	37.780782/-122.396559	37.743623/-122.454092	37.802327/-122.403298	37.800176/-122.439174	37.75087429/-122.42019
bikeasset005	37.775322/-122.438045	37.780519/-122.419014	37.783512/-122.440626	37.76408775/-122.41082	37.78091/-122.473486
bikeasset006	37.742786/-122.478151	37.777956/-122.406706	37.764631/-122.422973	37.78252/-122.409229	37.7865/-122.409517
bikeasset007	37.797088/-122.433485	37.768682/-122.420335	37.78285/-122.470532	37.772327/-122.412622	37.778623/-122.406247
bikeasset008	37.770641/-122.425406	37.78351336/-122.40081	37.751842/-122.429057	37.769995/-122.447179	37.751842/-122.429057
bikeasset009	37.76361/-122.481705	37.765062/-122.423249	37.777924/-122.400857	37.766196/-122.449777	37.73908/-122.414461
bikeasset010	37.784685/-122.40491	37.784685/-122.40491	37.7838/-122.452729	37.784685/-122.40491	37.73597064/-122.40613

Figure 4.2: Caption including a small piece of the whole csv file containing the measures of all the bikes per day

Health Flu Shot Location

Health Flu Shot Location carries data about the clinics offering flu vaccinations in the city. Each clinic gives information about the facility name and ID, address, contact number, and contact person, beginning date and time for the appointments, and some other important information such as cost and insurance details.


```
{
    'city': 'SanFrancisco',
    'begin_date': '2013-10-25T00: 00: 00',
    'end_date': '2013-10-25T00: 00: 00',
    'url': 'www.sfcdcp.org/fl',
    'street1': '2401KeithStreet',
    'facility_name': 'SoutheastHealthCenter',
    'facility_id': '938',
    'longitude': '-122.391891',
    'facility_type': 'HealthCenter',
    'phone': '(415)671-7000',
    'state': 'California',
    'contact': 'ErinM.Bachus, MPH',
    'postal_code': '94124',
    'end_time': '10: 00: 00-0500',
    'begin_time': '08: 00: 00-0500',
    'latitude': '37.72572',
    'country': 'UnitedStates',
```

```
    'eligibility': 'Adults18andOver',
    'notes': 'NoMedicareorInsuranceaccepted.',
    'currency_code': 'USD',
    'cost': '10'
}
```

Once again, this is more information than the one required, so only some fields will be used to generate the measures from the simulated sensor, that will give as well real time information about the number of appointments available in each center per day. The data modelling suposses slot times of 15 minutes per appointment, and considers an different occupancy percentages depending on the period of the year (in winter time, the vaccine is more requested than in summertime). Furthermore, an additional feature will be added as a complement to the information available: a connector with the website **Yelp** was prepared in this part, in order to offer ratings for each of the Health Centers as part of the measure.

doc_id	cost	health_center	rate	contact	position	availability	is_closed
docasset001	10	Chinatown Public Health Center	None	(415) 364-7934	37.797205/-122.411792	10	None
docasset002	10	Southeast Health Center	3.5	(415) 671-7000	37.72572/-122.391891	6	False
docasset003	10	Ocean Park Health Center	2.5	(415) 682-1950	37.762305/-122.482581	8	False
docasset004	10	Potrero Hill Health Center	2.0	(415) 648-3022	37.754/-122.398838	8	False
docasset005	10	Maxine Hall Health Center	3.0	(415) 292-1300	37.7823/-122.436084	8	False

Figure 4.3: Caption including a small piece of the whole csv file containing the measures of health centers offering flu vaccinations per day

Traffic Noise

Traffic Noise dataset encapsulates the information in a Shapefile format. On the one hand, a **shp** file contains a *shape*, which is the geographical area where data is collected; on the other hand, a **dbf** file contains the attributes, or data attached to the previous area. In order to mix them together, a script was prepared to input the shp and dbf files, and output information in JSON format altogether. Both files are coordinated and they contain the same number of samples.

```
{
    {
        'type': 'int',
        'name': 'Tract2010',
        'value': 16500
    },
    {
```

```
        'type': 'float',
        'name': 'Area_A',
        'value': [
            5996760.623304337,
            2109435.411795482,
            6000728.541410742,
            2110999.6630738676
        ]
    },
    {
        'type': 'float',
        'name': 'Area_smi',
        'value': [
            5996760.623304337,
            2109435.411795482,
            6000728.541410742,
            2110999.6630738676
        ]
    },
    {
        'type': 'string',
        'name': 'Tract2010_',
        'value': '016500'
    },
    {
        'type': 'float',
        'name': 'MAX',
        'value': '7.60000e+001'
    },
    {
        'type': 'float',
        'name': 'MEAN',
        'value': '4.60500e+001'
    }
}
```

In this case, as the information is already parsed into a JSON format, it is not necessary to load it into a CSV file. However, samples are too many and a logic will be applied to capture the required ones, based in a studio by the Urban Traffic Noise Policy in the city of Edmonton, Canada (around same population and dimensions of San Francisco, capital of the Alberta province, and fifth largest city in Canada), where a logarithmic representation of the typical daily traffic noise per hour in decibels will be the framework used to compare the measures obtained in San Francisco, in order to determinate which are the averaged places of noise, and the hot spots where the noise is greater. On top of that, the 311 complains to the city of San Francisco regarding high noise levels, will be a reference as well, included on the hot spot areas.

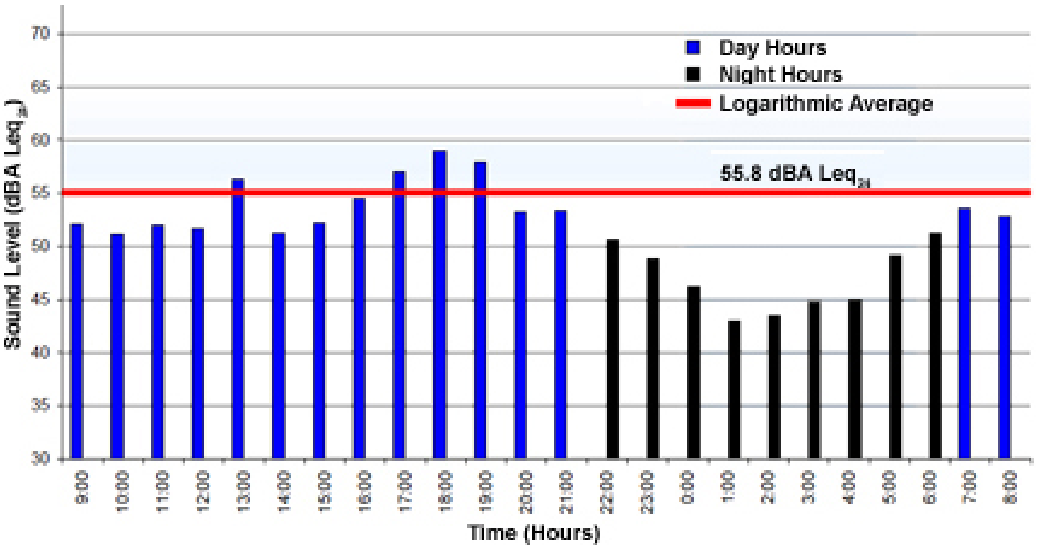

Figure 4.4: Daily average traffic noise levels per hour in a city of the size of San Francisco

In the Smart city representation, the average sound level per hour from the previous graph was the threshold to find the best locations of the city per hour, to be later depicted into a map. In order to find the closest measures to the desired ones, an algorithm was developed to sort the whole set according to the trigger value (minimum, maximum, or closer to a certain value), and filter a convenient number of samples.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
42	42	42	42	50	50	47	47	52	52	53	53	56	52	52	54	57	58	57	53	53	51	48	45

Table 4.3: Data model applied to the noises recorded on the city. Comparing the time (row in grey) in the day and the noise level measures (row in white)

Restaurant Scores

The Health Department has developed an inspection report and scoring system. After conducting an inspection of the facility, the Health Inspector calculates a score based on the violations observed. Violations can fall into:

- **High risk category:** Records specific violations that directly relate to the transmission of food borne illnesses, the adulteration of food products and the contamination of food-contact surfaces.

- **Moderate risk category:** Records specific violations that are of a moderate risk to the public health and safety.

- **Low risk category:** Records violations that are low risk or have no immediate risk to the public health and safety.

A file in CSV format contains the restaurant information, with address and location, plus an id matching with another CSV file that contains a score, date and type of inspection provided The score card issued by the inspection is up to date and available as well into this data set. In case any of the restaurants has suffered a violation, it is denoted by another CSV file referencing the restaurant id. A score legend with the marks is also available.

Scores	Description
0 - 70	Poor
71 - 85	Needs Improvement
86 - 90	Adequate
91 - 100	Good

Table 4.4: Meaning of each restaurant rate provided according to the inspection given

business_id	name	address	city	state	postal_code	latitude	longitude	phone_number
10	TIRAMISU KITCHEN	033 BELDEN PL	San Francisco	CA	94104	37.791.116	-122.403.816	
17	GEORGE'S COFFEE SHOP	2200 OAKDALE AVE	San Francisco	CA	94124	37.741.086	-122.401.737	14155531470
19	NRGIZE LIFESTYLE CAFE	1200 VAN NESS AVE; 3RD F	San Francisco	CA	94109	37.786.848	-122.421.547	
24	OMNI S.F. HOTEL - 2ND FLOOR PANTRY	500 CALIFORNIA ST; 2ND F	San Francisco	CA	94104	37.792.888	-122.403.135	
29	CHICO'S PIZZA	131 06TH ST	San Francisco	CA	94103	37.774.722	-122.406.761	14155251111
31	NORMAN'S ICE CREAM AND FREEZES	2801 LEAVENWORTH ST	San Francisco	CA	94133	37.807.155	-122.419.004	
45	CHARLIE'S DELI CAFE	3202 FOLSOM ST	San Francisco	CA	94110	37.747.114	-122.413.641	
48	ART'S CAFE	747 IRVING ST	San Francisco	CA	94122	37.764.013	-122.465.749	
50	SUSHI ZONE	1815 MARKET ST.	San Francisco	CA	94103	37.771.437	-122.423.892	14155621114

Figure 4.5: Caption of the CSV file containing information regarding to the restaurants

business_id	score	date	type
10	94	20140729	routine
10	92	20140114	routine
10	98	20121114	routine
17	94	20140312	routine
17	94	20130605	routine
17	100	20120823	routine
19	94	20141110	routine
19	94	20140214	routine

Figure 4.6: Caption of the CSV file containing rates about the issued restaurants, and date of the inspecion

business_id	date	description
19	20141110	Improper storage of equipment utensils or linens
19	20141110	Inadequate food safety knowledge or lack of certified food safety manager
19	20140214	Permit license or inspection report not posted [date violation corrected: 11/10/2014]
24	20140612	Improper storage of equipment utensils or linens [date violation corrected: 11/24/2014]
29	20140818	Inadequate procedures or records for time as a public health control [date violation corrected: 8/20/2014]
10	20121114	Unclean or degraded floors walls or ceilings [date violation corrected: 1/14/2014]
10	20140729	Unapproved or unmaintained equipment or utensils [date violation corrected: 8/7/2014]
10	20140729	Insufficient hot water or running water [date violation corrected: 8/7/2014]
10	20140114	Improper storage of equipment utensils or linens [date violation corrected: 1/24/2014]
17	20130605	Unclean or degraded floors walls or ceilings [date violation corrected: 3/12/2014]

Figure 4.7: Caption of the CSV file containing the violation description regarding to the inspection

Since the information contains historical review of the inspections and restaurants, and duplicated data, it was necessary to refactor it all, presenting the following result also in CSV format, containing information about the venues, score and issue description in case the issue is still open.

id	name	address	city	state	zip	latitude	longitude	phone_numl	score	description
10	TIRAMISU KI	033 BELDEN I	San Francisc	CA	94104	37.791.116	-122.403.816		94	
17	GEORGE'S CC	2200 OAKDA	San Francisc	CA	94124	37.741.086	-122.401.737	1,4156E+10	94	
19	NRGIZE LIFES	1200 VAN NE	San Francisc	CA	94109	37.786.848	-122.421.547		94	Inadequate food safety knowledge or lack of certified food safety manager
24	OMNI S.F. H(	500 CALIFOR	San Francisc	CA	94104	37.792.888	-122.403.135		96	Inadequate and inaccessible handwashing facilities
29	CHICO'S PIZZ	131 06TH ST	San Francisc	CA	94103	37.774.722	-122.406.761	1,4155E+10	86	
31	NORMAN'S I	2801 LEAVEN	San Francisc	CA	94133	37.807.155	-122.419.004		100	
45	CHARLIE'S DE	3202 FOLSON	San Francisc	CA	94110	37.747.114	-122.413.641		94	Wiping cloths not clean or properly stored or inadequate sanitizer
48	ART'S CAFE	747 IRVING S	San Francisc	CA	94122	37.764.013	-122.465.749		92	Foods not protected from contamination
50	SUSHI ZONE	1815 MARKE	San Francisc	CA	94103	37.771.437	-122.423.892	1,4156E+10	100	
54	RHODA GOLI	2180 POST S1	San Francisc	CA	94115	37.784.626	-122.437.734	1,4155E+10	100	
56	CAFE X + O	1799 CHURCH	San Francisc	CA	94131	37.742.325	-122.426.476	1,4156E+10	98	Permit license or inspection report not posted

Figure 4.8: Caption of the CSV file containing merged information from the previous, filtering the last inspection per restaurant

4.1.2 Deployment of an IoT platform

Like it was previously explained, FIWARE Service Hosting Providers offer cloud image services, that manage pre-built images that can be used to provision VMs or Linux containers, typically comprising only the files of the application and its dependencies.

Amazon Web Services was the Hosting Server selected for this project, since FIWARE Platform is deployed there, and it can be easily instantiated from there as well. Inside this service, three Amazon Machine Images (AMIs) built from a IoT Fiware Stack are available: Secured IoT Stack v0.1.6, including *Context Broker*, *Portal*, *STH*, *Access Control* and *PEP*; Secured IoT Stack v0.1.7, including *Context Broker*, *Portal*, *STH*, *Access Control* and *PEP*; and Basic IoT Stack v0.2.0, including *Context Broker* and *IoT Agents* listening HTTP UL2.0 and MQTT protocols. These AMIs can be seen as real FIWARE bundles ready to deploy, each of them containing a set of components that share common goals towards data.

Despite IoT Stack v0.1.6 and v0.1.7 fit closer with the whole FIWARE platform, due to the requirements of the application, an *IoT Agent* is need to fulfill the simulation of data depicted as real sensor information. For that reason, the AMI deployed for this project is the IoT Stack v0.2.0. After choosing the AMI that will be deployed, the details of the instance launched have to be configured.

<table>
<tr><td>Instance Name</td><td colspan="4">i-3333f38f (randomly chosen)</td></tr>
<tr><td>Instance Type</td><td colspan="4">t2.micro (the free tier choice)</td></tr>
<tr><td>Memory</td><td colspan="4">1 GB (the free tier choice)</td></tr>
<tr><td>Network Performance</td><td colspan="4">Low to Moderate (the free tier choice)</td></tr>
<tr><td rowspan="4">Security Group Details</td><td>Type</td><td>Protocol</td><td>Port</td><td>Range Source</td></tr>
<tr><td>SSH</td><td>TCP</td><td>22</td><td>0.0.0.0/0</td></tr>
<tr><td>Custom</td><td>TCP</td><td>1026</td><td>0.0.0.0/0</td></tr>
<tr><td>Custom</td><td>TCP</td><td>8080-8086</td><td>0.0.0.0/0</td></tr>
</table>

Table 4.5: Configuration parameters of the AWS instance

Once these details are sorted, a PEM file (Privacy Enhanced Mail Security Certificate) is generated containing the certificates of the instance deployed, and it is required for the access via SSH to the machines of the instance. Also, a public IP is associated with the instance, serving as well as the API REST to access the platform. This key changes everytime the machine is stopped and restarted. The security group opens the desired ports, in order to access the IoT Platform components through them.

Port designed to access *Orion Context Broker*	1026
Port designed to access *IoT Agents*	8080
Port designed to access *IoT Agent Managers*	8086

Table 4.6: List of open ports and assigned components

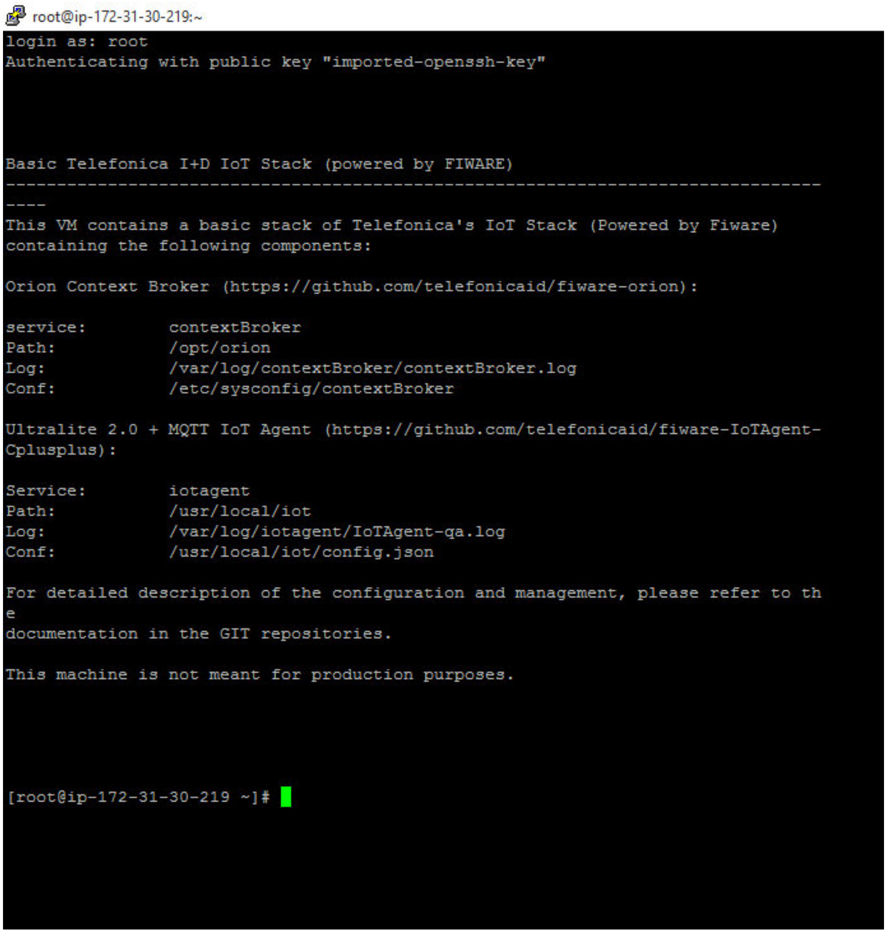

Figure 4.9: SSH Connection to the instance as root server

4.1.3 Management of an IoT Platform

In order to access the platform, the data captured from the chosen datasets will be converted into real measures from sensors, that will be sent using a light protocol to an *IoT Agent* able to listen that protocol, and also enabled to communicate with *Context Broker*.

IoT Agents provide a series of rules in order to accept information and later push it to *Context Broker*. These steps can be found in their GitHub repository, configuration API section, and they are going to be followed here to assure a successful and reliable transmission of data.

Creating a service

Attending to the security accesses to the FIWARE IoT platform, before sending any information, it is necessary to adopt a multitenant structure for the data collection that will be published inside *Orion Context Broker*, that will serve as well to keep data separated all the time, avoiding misleading information at the end. There are two tenancy levels available to provide a FIWARE service: FIWARE service, and FIWARE service path. The fact that all data belongs to the same city, makes clear that at least one of the levels should hold it all, and for that reason, the first tenancy level will be common. Based in San Francisco, the Smart City service in this project will be called **smartfrisco**

However, the different nature of the data sources would complicate the access and mess the inner organization if different sensor types mix together. For that reason, the second tenancy level will difference the 5 data types associated with each of the basic features that compose the Smart City, calling them with a characteristic name to the area where they belong, and a slash symbol at the head, because it is compulsory in this tenancy level specification.

Tenancy level	Tenant label	Tenant Name
1st	Fiware-Service	smartfrisco
2nd	Fiware-ServicePath	/bicycles /healthcare /noisetraffic /restrates

Table 4.7: Tenancy levels designed for this project

To add a service and subservice, an HTTP post against the *IoT Agent* is made through the request that follows this explanation, where *apikey* and *token* are unique for each subgroup, the *entity type* will locate the device later when it is posted in *Context Broker*, and the *resource* indicates the protocol being used, to indicate that the *IoT Agent* is able to listen it. This project will use UL2.0 as reference procotol, thus, resource will be always the same: "/iot/d". Down below, the services provided for each data set are specified.

- Service provided for Bicycle Parking dataset:

```
POST http://<INSTANCE_HOST>:8080/iot/services

HEADERS:
Fiware-Service: smartfrisco
Fiware-ServicePath: bicycles

PAYLOAD:
{
    "services": [
        {
            "apikey": "4p1k3y",
            "token": "150196",
            "cbroker": "http://<INST_HOST>:1026",
            "entity_type": "/bicycles",
            "resource": "/iot/d"
        }
    ]
}
```

- Service provided for Healthcare Centers dataset:

```
POST http://<INSTANCE_HOST>:8080/iot/services

HEADERS:
Fiware-Service: smartfrisco
Fiware-ServicePath: healthcare

PAYLOAD:
{
    "services": [
        {
            "apikey": "h34lzk3y",
            "token": "240291",
```

```
                "cbroker": "http://<INST_HOST>:1026",
                "entity_type": "/healthcare",
                "resource": "/iot/d"
            }
        ]
    }
```

- Service provided for Noise Traffic dataset:

```
POST http://<INSTANCE_HOST>:8080/iot/services

HEADERS:
Fiware-Service: smartfrisco
Fiware-ServicePath: noisetraffic

PAYLOAD:
{
    "services": [
        {
            "apikey": "n0oz3k3y",
            "token": "100363",
            "cbroker": "http://<INST_HOST>:1026",
            "entity_type": "/noisetraffic",
            "resource": "/iot/d"
        }
    ]
}
```

- Service provided for Restaurant rates and inspections dataset:

```
POST http://<INSTANCE_HOST>:8080/iot/services

HEADERS:
Fiware-Service: smartfrisco
Fiware-ServicePath: /restrates

PAYLOAD:
{
    "services": [
        {
            "apikey": "3At4udk3y",
            "token": "180166",
```

```
            "cbroker": "http://<INST_HOST>:1026",
            "entity_type": "restrates",
            "resource": "/iot/d"
        }
    ]
}
```

Later on, the meaning of the parameters pased to create a service will be really explained, however some of them are not used, as the *token*, that would be required for Trust Token security access, and this security level is not required to interact with *Context Broker*. When a service is created, the response from the *IoT Agent* is as follows:

```
HTTP/1.1 201 Created
Connection: Keep-Alive
Host: <INSTANCE_HOST>:8080
Content-Length: 0
```

On the opposite case, when a service has not successfully been created, a *404* response would appear.

Creating a device

When communication is launched towards *Context Broker*, there are two ways of sending information from the *IoT Agent*. On the one hand, devices can be provisioned first to modify the entity conditions that will be contained in *Context Broker*. On the other hand, if measures are sent to the *IoT Agent* wihout a previous provision of the device, this component will transform the measure contents as an entity whose name will be the name of the device.

Despite the second choice is easier, as the whole step of creating a device would be no longer required, several features as renaming the names of the attributes, or posting location type measures, would not be possible. For that reason, all the devices of this project were provisioned before sending them to *Context Broker*. Once again, the way to provision devices is made through an HTTP POST request. But before going through it, let's analyse what a device should contain to be properly provisioned, helped by the table below.

Parameter name	Parameter example	Parameter description
device id	device001	Name of the device provisioned. Must be unique
entity name	my_entity	Name of the entity in *Context Broker*. Must be unique
timezone	Pacific/California	Irrelevant for this project, but compulsory to create the device
attributes object id name type	 t temperature int	 short name at the measure real name in Context Broker Types: int, float, string...
static attributes name type value	 location string WGS84	This is not useful unless the attribute is a location attribute, in that case it follows this example

Table 4.8: Parameters required to fill in order to prepare a device provision

When static attributes carry the information specified in the table, a location type measure is expected. Location measures, as it will be later explained, contain two float numbers separated by a slash symbol, that will be understood as *latitude, longitude* in *Context Broker*.

An example of a device provisioned for the 5 types of sensors worked out in this project is as follows:

- Device provisioned for the Bicycle Parking dataset:

```
POST http://<INSTANCE HOST>:8080/iot/devices

HEADERS:
 Fiware-Service: smartfrisco
 Fiware-ServicePath: /bicycles

PAYLOAD:
{
    "devices": [
        {
            "attributes": [
                {
                    "name": "position",
                    "object_id": "1",
                    "type": "coords"
```

```
                }
            ],
            "device_id": "bikeasset001",
            "entity_name": "bicycle1",
            "entity_type": "bicycles",
            "static_attributes": [
                {
                    "name": "location",
                    "type": "string",
                    "value": "WGS84"
                }
            ]
        }
    ]
}
```


- Device provisioned for the Healthcare Centers dataset:

```
POST http://<INSTANCE HOST>:8080/iot/devices

HEADERS:
 Fiware-Service: smartfrisco
 Fiware-ServicePath: /healthcare

PAYLOAD:
{
    "devices": [
        {
            "attributes": [
                {
                    "name": "position",
                    "object_id": "l",
                    "type": "coords"
                },
                {
                    "name": "cost",
                    "object_id": "cs",
                    "type": "float"
                },
                {
                    "name": "rate",
```

```
                "object_id": "r",
                "type": "float"
            },
            {
                "name": "availability",
                "object_id": "av",
                "type": "int"
            },
            {
                "name": "is_closed",
                "object_id": "ic",
                "type": "bool"
            }
        ],
        "device_id": "docasset001",
        "entity_name": "health_center1",
        "entity_type": "healthcare",
        "static_attributes": [
            {
                "name": "location",
                "type": "string",
                "value": "WGS84"
            }
        ]
    }
  ]
}
```


- Device provisioned for the Noise Traffic dataset:

```
POST http://<INSTANCE HOST>:8080/iot/devices

HEADERS:
 Fiware-Service: smartfrisco
 Fiware-ServicePath: /noisetraffic

PAYLOAD:
{
    "devices": [
        {
            "attributes": [
```

```
                    {
                        "name": "position",
                        "object_id": "l",
                        "type": "coords"
                    },
                    {
                        "name": "max_noise",
                        "object_id": "mn",
                        "type": "float"
                    },
                    {
                        "name": "avg_noise",
                        "object_id": "an",
                        "type": "string"
                    }
                ],
                "device_id": "noiseasset001",
                "entity_name": "noise_sensor1",
                "entity_type": "noisetraffic",
                "static_attributes": [
                    {
                        "name": "location",
                        "type": "string",
                        "value": "WGS84"
                    }
                ]
            }
        ]
    }
```


- Device provisioned for the Restaurant rates and inspections dataset:

```
POST http://<INSTANCE HOST>:8080/iot/devices

HEADERS:
 Fiware-Service: smartfrisco
 Fiware-ServicePath: /restrates

PAYLOAD:
{
    "devices": [
```

```
        {
            "attributes": [
                {
                    "name": "position",
                    "object_id": "1",
                    "type": "coords"
                },
                {
                    "name": "score",
                    "object_id": "sc",
                    "type": "int"
                }
                {
                    "name": "description",
                    "object_id": "dcp",
                    "type": "string"
                }
            ],
            "device_id": "restasset001",
            "entity_name": "restaurant_name1",
            "entity_type": "restrates",
            "static_attributes": [
                {
                    "name": "location",
                    "type": "string",
                    "value": "WGS84"
                }
            ]
        }
    ]
}
```

Thus, when a measure is sent to the *IoT Agent*, it will be matched with the previously provisioned entity pattern. When a device is created, the response from the *IoT Agent* is as follows:

```
HTTP/1.1 201 Created
Connection: Keep-Alive
Host: <INSTANCE_HOST>:8080
Content-Length: 0
Location: /iot/devices/device_id
```

On the opposite case, when a device has not successfully been created, a *404* response would appear.

Sending a measure

In the previous step of transforming data into real sensors, the key goal was preparing real devices, or assets, with several measures stored inside. This is important, because the protocol that will be used to transmit measures is **Ultralight 2.0**, that works via HTTP, sending POST requests with the following url:

```
POST http://<HOST>:<PORT>/?k=<api-key>&i=<deviceID>
&t=<timestamp>&ip=<command_URL>&getCmd=1

Body: <measures_data>
```

Query parameters *k*, and *i* are essential, and the rest are optional (*ip* and *getCmd* are only used when receiving command measures, as ACK notification). Payload body separates attributes of the measure in *alias* and *value*, where *alias* denotes the device sensor identifier specified in the model and *value* contains the device sensor measure value, with a pipe symbol. Several attributes can be sent together, concatenated with a hash symbol. Also, *timestamp* can be sent in the payload, introduced before the *alias*. Furthermore, in case of sending location measures, those follow the convention *latitude/longitude*.

```
Sending one measure: <alias>|<body>

Sending a location: <alias>|<latitude>/<longitude>

Sending with timestamp: <timestamp>|<alias>|<body>

Sending several measures: <alias1>|<body1>#<alias2>|
                          <body2>#...
```

Once again, let's take a look to the different measures sent in each of the datasets handled previously:

- Measure sent from a Bicycle Sensor.

  ```
  GET http:/<INSTANCE_HOST>:8085/iot/d?i=bikeasset001
      &k=4p1k3y&t=2015-08-19T4:00:00Z&d=l|37.78/-122.39
  ```

  In the case of the Bicycle Sensors, one of this measures would be produced each 30 minutes and sent to the platform.

- Measure sent from a Health center Sensor. Since there are more measures now, a POST request was made to avoid a messy url call.

  ```
  POST http:/<INSTANCE_HOST>:8085/iot/d?i=docasset001
      &k=h34lzk3y&t=2015-08-19T

  PAYLOAD:
  1|37.762305;-122.482581#av|8#c|10#r|2.5#ic|False
  ```

- Measure sent from a Traffic Noise Sensor.

  ```
  POST http:/<INSTANCE_HOST>:8085/iot/d?i=noiseasset001
      &k=n0oz3k3y&t=2015-08-23T1:00:00Z

  PAYLOAD:
  1|37.743359/-122.474291#mn|76#an|38.3
  ```

  In this case, one of this measures is sent every hour to all the assets in the network.

- Measure sent from a Restaurant Sensor.

  ```
  POST http:/<INSTANCE_HOST>:8085/iot/d?i=restasset001
      &k=3At4udk3y&t=2015-08-23T15:00:00Z

  PAYLOAD:
  1|37.743359/-122.474291#sc|92#dep|Improper food storage
  ```

  In this case, the measure is sent on time when a new inspection report is done.

When a measure is sent, the automatic response from the *IoT Agent* is *200OK*, even if the measure hasn't been properly sent. On top of that, there is not any ACK response delivered to the sensor if something went wrong. For that reason, it is important to make sure that the measure has arrived OK, by querying from time to time to *Context Broker*.

To sum up, the following table contains a brief review of all the configurations by provision or southbound API calls, that have been exposed previously in this chapter.

Service	Device provided	Measures contained
Service-Path: bicycles Type: bicycles Apikey: 4p1k3y Token: 150196	Name: bikeasset<num> Entity Name: bicycle<num> Attributes: position	l (position) Timestamp
Service-Path: healthcare Type: healthcare Apikey: h34lzk3y Token: 240291	Name: docasset<num> Entity Name: health_center<num> Attributes: position, cost, rate, availability, is_closed	l (position) c (cost) r (from rate) av (available slots) ic (is_closed) Timestamp
Service-Path: noisetraffic Type: noisetraffic Apikey: n0oz3k3y Token: 100363	Name: noiseasset<num> Entity Name: noise_sensor<num> Attributes: position, max_noise, avg_noise	l (position) mn (max_noise) an (avg_noise) Timestamp
Service-Path: restrates Type: restrates Apikey: 3At4udk3y Token: 180166	Name: restasset<num> Entity Name: restaurant_name<num> Attributes: position, score, description	l (position) sc (score) dep (description) Timestamp

Table 4.9: Table as a summary of the whole set of operations performed for this project, and the variable names given to all the data sets

Chapter 5

Tests and results

As a result of the previous development within the FIWARE platform, several testing challenges have been prepared to try the platform and its capabilities. This chapter offers a detailed description of the tests provided, and the results of these tests.

- Data visualization.

  Extrapolate the published data somewhere else is basic to proof one of the major features of the platform: the adaptability provided for the information retained inside. When the platform intends to be a common framework for many clients, that would need their datasources to be easily and quickly accessible from any place, the platform should be fully integrable with a certain third party endpoint. An example of a third party tool that will usually be required when interacting with the platform is a data visualization tool, specially considering that the scenario where this project is framed is a Smart City, and Smart City clients are citizens that need easy and visual tools to view data.

- Integration tests.

  A good communication among the inner components of the platform is vital to show a reluctant platform that is capable to offer a common service despite it is a modular structure. And the best way to expose the interaction between the components is via integration testing tools. By preparing specialized dataflows that test the paths communicating the different components, a real vision of the reliability of the platform is obtained, of course, when tests are successfully passed.

5.1 Data visualization

In order to show the results of the published data in Context Broker, a dashboard was chosen because data visualized results are simpler to understand and more eye-catching than a bunch of JSON formatted files.

The visualization tool chosen was *Freeboard* because, as its own name indicates, it is a free tool. It is a open source real-time dashboard builder for web mashups, handy for IoT applications. It provides a plugin architecture for creating datasources, which fetch data; and widgets, which display data. *Freeboard* then does all the work to connect the two together. Another feature of this tool is its ability to run entirely in the browser as a single-page static web app without the need for a server. The feature makes it extremely attractive as a front-end for embedded devices which may have limited ability to serve complex and dynamic web pages.

In the case of the FIWARE IoT platform, *Orion Context Broker* was the datasource, and several widgets were used to display the information inside. This section describes the steps followed to encompass the available data with the *Freeboard* requirements.

Freeboard as context Consumer

This concept was previously introduced when talking in depth about *Context Broker*. In this case, *Context Broker* contains the information published inside, and the way of taking it outside is via *queryContext* operations. But, since the amount of requests to access all the available data is huge, *Freeboard* is registered in Context Broker as Context Consumer for all the data published in the demanded service, receiving information updates based *on time* or *on change* (depending on the details of the subscription with *Freeboard*, in the case of this project, *on time*). Figure 5.1, obtained from the User and Programmers Guide of *Orion Context Broker* [40], exposes how these two bodies communicate.

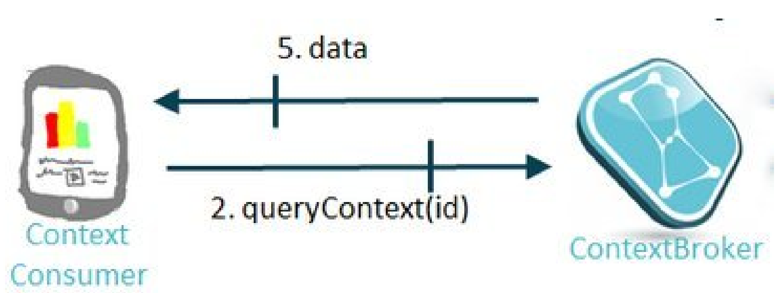

Figure 5.1: Example on how freeboard communicates with Context Broker

Context Broker plugin in Freeboard

Freeboard is open source, thus any imaginable plugin can be switched for opening new datasources into the dashboard tool. *Orion Context Broker*, as the major exponent of one of the most complex IoT platforms existing nowadays, has of course its own plugin available for instantaneously switch data on it. The following figure (Figure 5.2) exposes this plugin, available in *Freeboard* website [51].

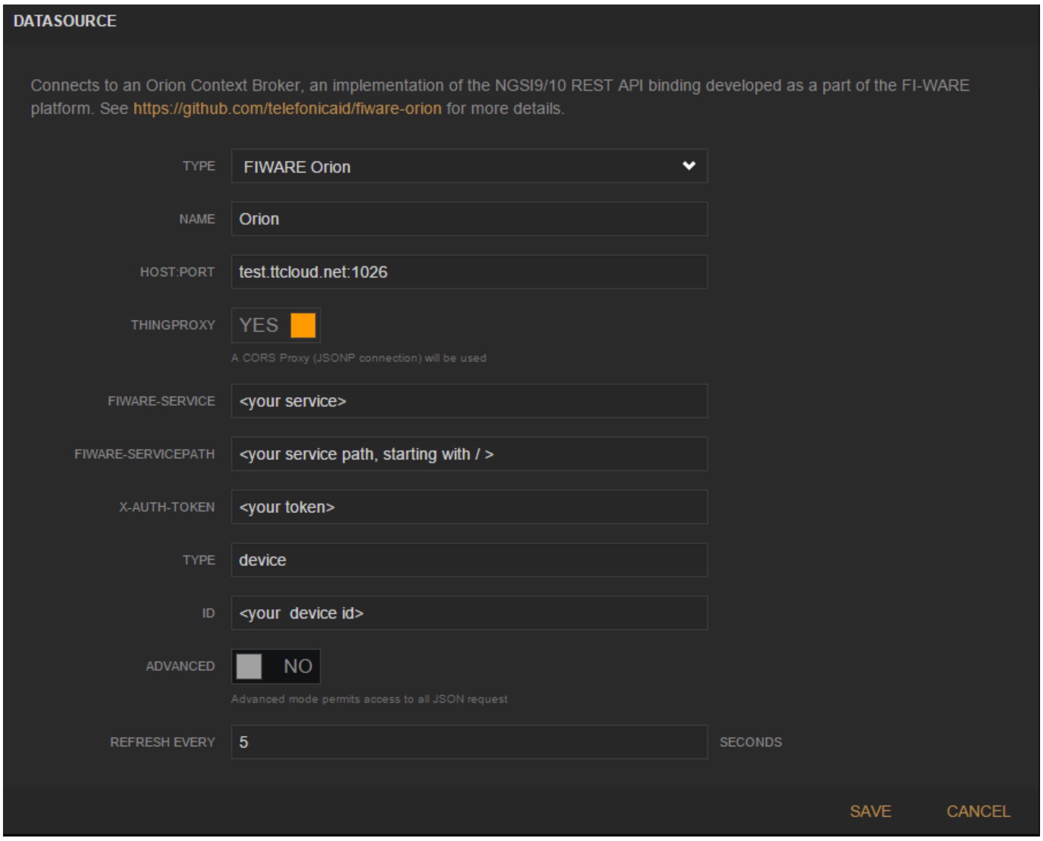

Figure 5.2: Context Broker plugin available in Freeboard

Preparing widgets

Widgets in freeboard are shaped into small panels that offer Google Maps depicted maps, pictures, HTML incrusted items, text, indicator lights, etc.

For this project, five different dashboards associated with the five different data handled and published in Context Broker, were created. At the end of this essay, Appendix B takes a wider look through graphical results.

5.2 Integration tests

It is the job of a QA (Quality Assurance) engineer to assure the status of the platform and perform the pertinent tests and validations to confirm reliability. Among other quality tests developed by the QA, Integration testing validates the involvement of all the components to make a reliable system. It is the phase in software testing in which individual software modules are combined and tested as a group. It takes as input one of the modules to be tested, and following the flow desired, applies tests defined in an integration test plan, to finally take as output the last module of the integrated system in the validation. Integration tests described in this part follow a botton up testing, where the first component tested is the lowest level component: the *IoT Agent*, and it will follow a flow that goes to the *Context Broker* where datasets in the flow will be validated.

5.2.1 BDD practise

Behavior Driven Development was be the Agile methodology applied in the testing process, defining behavioural specifications that indicates how the desired behaviour should be specified, as a group of user stories that follow a structure similar to the following:

- **Title** of the story.
  - **Who** is the driver of the story, who performs the whole action.
  - **What** effect is the actor contributing to.
  - **What** actions will derive from the former effect.
- **Scenarios** or acceptance criteria, describing specific cases of the story.
  - **Given** an initial condition assumed for the actor.
  - **States** event triggers when initial conditions change as they flow over the path integrated.
  - **Then** it states the expected outcome.

Chasing the previous behaviour, individual scenarios were defined where several types of data that could be possibly shared among components, and tested using the agile testing software development technique *Behave*, that runs tests developed with BDD and backed up by *python* programming language.

5.2.2 Defining scenarios

Before start defining the scenarios, it is important to think about the previous chapter and how data was transmitted through the platform. Basically, there were data flows to test:

- Single measures sent to the platform, like in the case of **bicycle** assets.
- Multiple measures sent to the platform, like in the rest of the cases.
- Attempt to send data to other subservices, to test how independent is the second tenancy level, where a service/subservice set cannot listen to the rest of them.

Scenario 1: Single measures

The BDD ontology designed for this case would be as follows.

```
Scenario Outline: Data sent from sensors to IOTAgent should be stored in CB
    Given I add a Service "<SERVICE>" to the IOTAgent
    And I add a Device "<DEVICE>" in that IOTAgent service
    And I wait 1 seconds
    When a measure with key, value "<MEAS_KEY>", "<MEAS_VALUE>" is sent
        from the device "<DEVICE>" with the apikey "<APIKEY>"
    Then the last value in CB for the entity ID "<ENT_NAME>" and type
        "<ENT_TYPE>" should match with the sent measure

Examples:
| SERVICE     | DEVICE| APIKEY| MEAS_KEY|MEAS_VALUE| ENT_NAME| ENT_TYPE|
| smartfrisco| dev1  | apikey| c       |  green   | room1   | thing   |
| smartfrisco| dev2  | apikey| t       |  6.97    | room2   | thing   |
| smartfrisco| dev3  | apikey| l       |  45/-34  | room3   | thing   |
| smartfrisco| dev4  | apikey| B       |  False   | room4   | thing   |
```

In this scenario, a single measure was sent to the service *smartfrisco*, containing a *string* type measure, a *float* type measure, a *location* type measure, and a *boolean* type measure. All the measures were sent to the same subservice, designed just for this exercise: */testsubservice*.

Scenario 2: Multiple measures

The BDD ontology designed for this case would be as follows.

```
Scenario Outline: Data sent from sensors to IOTAgent should be stored in CB
    Given I add a Service "<SERVICE>" to the IOTAgent
    And I add a Device "<DEVICE>" in that IOTAgent service
    And I wait 1 seconds
    When a measure with key, value "<MEAS_KEY>", "<MEAS_VALUE>" is sent
        from the device "<DEVICE>" with the apikey "<APIKEY>"
    Then the last value in CB for the entity ID "<ENT_NAME>" and type
        "<ENT_TYPE>" should match with the sent measure

Examples:
```

```
| SERVICE      | DEVICE|APIKEY| MEAS_KEY|MEAS_VALUE          |ENT_NAME| ENT_TYPE|
| smartfrisco| dev5   |apikey| c;ack    |green;True          |room5    | thing    |
| smartfrisco| dev6   |apikey| t;p;l    |6.97;722;122/-32 |room6    | thing    |
```

In this scenario, a couple flows containing several measures each was sent. The second example is important, because it contains a *location* type measure, and some more, and it confirms that location can be sent as one attribute more. All the measures were sent to the same subservice, designed just for this exercise: */testsubservice*.

```
Scenario Outline: Data sent to an existing subservice
    Given I add a Service "<SERVICE>" to the IOTAgent
    And I add a Device "<DEVICE>" in that IOTAgent service
    And I wait 1 seconds
    When a measure with key, value "<MEAS_KEY>", "<MEAS_VALUE>" is sent
        from the device "<DEVICE>" with the apikey "<APIKEY>"
    Then I receive a 404 status code from Context Broker.

Examples:
| SERVICE      | DEVICE|APIKEY  | MEAS_KEY|MEAS_VALUE|ENT_NAME| ENT_TYPE|
| smartfrisco| dev7   |apikey1| x        |25         |room5    | thing    |
```

In this scenario, only one flow was tested, trying to create a device but retrieving it from another subservice. If subservices are idle, no device should be found under those condition. The measure was sent to a subservice designed just for this exercise: */testsubservice1*.

After performing the whole set of tests, since entities and devices remain in the memory pools of each component, clean up process was made to remove these testing suites.

Chapter 6

Conclusions and future research

The previous testing exercises confirm that the IoT platform provided by FIWARE is suitable for Smart City development, being able to publish many data types completely independent from each other due to the multitenancy security access, as integration tests have previously proved.

Adaptability with a third party site was also a success, as it can be exposed in the data visualization results, that were developed straight forward after data was published in the platform due to the plugin available in the dashboard domain chosen.

Finally, reliability meets the expected conditions with a successful pass of the integration tests submitted, exposing a good relationship between all the components in the platform, which despite of presenting a modular solution, brings fast communication calls among them. For that reason, FIWARE platform can evolve as a open source platform characterized by embracing many modules from many vendors, provided that the backend environment is fast and reliable.

However, the more tricky issue when accessing the platform, is the verbosity found in the requests. The ontological structure is quite good and intuitive, but facing real devices, sometimes it is not easy to adapt their conditions to the ones expected within the platform. Also, the fact that sending measures offers always a successful response no matter whether measures have been stored in the platform or not, makes a little bit hard to think about applications where data is really important, such as eHealth scenarios or industrial IoT with relevant data.

Nevertheless, approaching this platform and deploy it has been easy in general terms, and this can be an advantage when attracting many types of public to the platform: from freelance develops or students, to SMEs, start-ups or big companies, that can find their spot inside of the versatility offered by FIWARE platform. Indeed, this is a small project with a small scope and resources, but with the right tools, really big scenarios under the framework of an Smart City can be provided due to the FIWARE platform.

6.1 Future research

Future research in this project

This project takes the scope of the Smart City development and uses several components of the FIWARE IoT platform to approach it. Using some other components would give to the application developed for this project a richer variety of services provided. For instance, using the *CEP* component to create rules when measures are published in *Context Broker*: if a noise sensor captures a very high level of decibels, a notice could be sent to the Police Department to check the area where the device is. Or plugging the bicycles data set to a business intelligence tool, a prediction about which parking slots are more frequented would help the city management, that could add some extra slots in that parking.

Future research in the Platform

The more the platform is able to grow, the higher the necessity will be for developing more *IoT Agents* to communicate with all the southbound protocols promoted by vendors in hardware devices. In terms of sotware, *IoT Agents* could offer a more reliable method when they are attacked by southbound APIs (ie. when a measure is received), such as ACKs delivered back to the device to assure a sucessful transmission, or a wider HTTP response with rejection clause in case a measure has not been received with the proper format, or the rest of the platform rejects the request.

The availability of third party connectors could grow as well with the development of the platform. Currently, connectors for map visualization, a simple dashboard, or relational and non relational databases are available, but some more could be necessary, such as voice recognition interface tools.

Future research in Smart Cities

Future visions and projects in the Smart City development put its efforts into Energy research challenges. According to *Nielsen* [53], *The Smart Energy City is highly energy and resource efficient, and is increasingly powered by renewable energy sources; it relies on integrated and resilient resource systems, as well as insight-driven and innovative approaches to strategic planning. The application of information and communication technologies is commonly* *a means to meet these objectives.* Energy research is the next target research topic in Smart Cities, focused in fields as Smart Electricity Grids, Smart district Heating and Cooling, and Energy saving by technological means.

But energy is not the only door open for research in Smart Cities: location based applications, present in Smart Transportation or WiFi positioning systems, are also a target for possible researchers, since *always on* is another of the key concepts of a smart city, where clients and servers are requested to be always available. Location based applications will be the framework in which services such as location based automotive navigation, fleet tracking or real-time parking will born and raise in the next few years.

Gartner Inc, the vendor Ratings Resarch Methodology has built a graph to discuss when new technologies will become a reality, whether some of them will succeed or fail, or which ones will be more commercially viable. Figure 6.1, from the Gartner Hype Circle 2012 [54], gives a glimpse of Smart Cities in the near future.

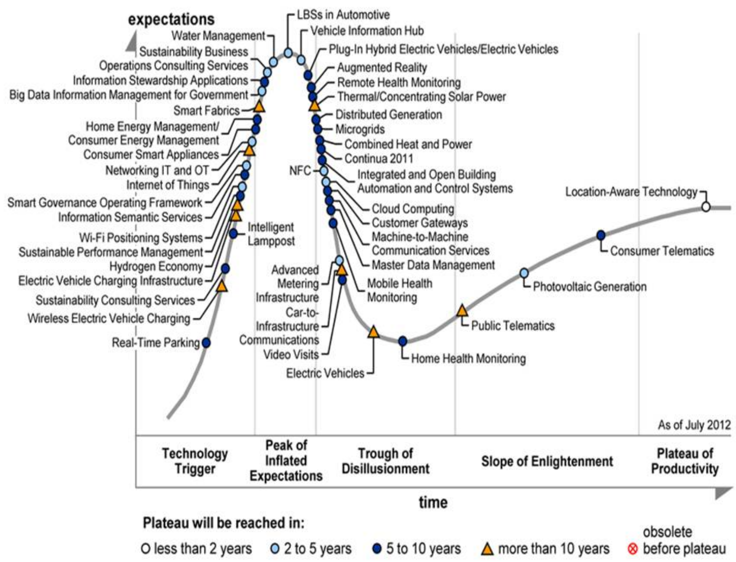

Figure 6.1: Gartner Hype Circle for Smart City Technologies, 2012

Chapter 7

Costs and management of the project

In this chapter, a detailed list of the different costs attached to the development of this project will be introduced, inside a portfolio statement sheet. Also, connected to the strong customer orientation of this project, an estimation outcomes balanced with the costs of the suitable level of production will be provided.

Finally, in the management section of this chapter, different project schedules that illustrate the whole process of this essay will be depicted in charts to evaluate the whole development process in a graphical way.

7.1 Budget and costs

The whole amount of expenses will be separated in two different groups, **direct costs** and **indirect costs**.

- Direct costs: Those including materials, machinery, or human resources involved (salaries). All these costs assume the greater weight of the project's budget.
- Indirect costs: Those including renting expenses or utilities. These costs are around the 20% of the direct costs.

It is also necessary to take into account that the quantities are expressed in Euro, and they will only consider two decimal digits, that will be rounded up in case of need.

7.1.1 Portfolio Budget Statement

This section offers a detailed calculation to estimate the whole budget, plus a final portfolio budget statement containing the cost sheet. Costs are updated to the new tax regulation applicable from January 1st 2015.

Material costs included

- Laptop: The computer designed for this project was the HP Pavilion dv6 Notebook PC, from the vendor Hewlett-Packard.

Memory	Processor	Clock Speed	System type
8GB RAM	Intel Core i5-2450M	2.5GHz	64b

Table 7.1: Details of the Laptop used

- Router WiFi purchased with the Internet installation to ONO Cableuropa S.A., from the vendor Thompson, model TWG870.
- 22 inch monitor: Within this project, an extra screen was used for code development Equipment details: ViewSonic VA2212M-LED 22-Inch LED-Lit Monitor.

Personal costs included

- Counselors salary: Salary designed to the tutors of this project, Marcos Reyes Ureña as I+D Director and Victor Pedro Gil Jimenez as Quality Engineer. Wages have been extracted from the Study of Remuneration of 2014.

$$\frac{65000}{12} = 5416.67 \text{ €}/month \quad \frac{5416.67}{4 \cdot 5 \cdot 8} = 34 \text{ €}/hour$$

  number of hours for the I+D Director: 54 $\quad 34 \cdot 54 = 1836$

  number of hours for the Quality Engineer: 10.8 $\quad 34 \cdot 10.8 = 367.2$€

- Junior Engineer Salary: Salary designed to the developer Andrea Gonzalez Mallo, as I+D Junior Engineer.

$$\frac{27000}{12} = 2250.00\text{€}/month \quad \frac{5416.67}{4 \cdot 5 \cdot 8} = 14 \text{ €}/hour$$

  number of hours for the I+D Junior Engineer: 360 $\quad 14 \cdot 360 = 5040$€

7.1.2 Portfolio Cost Sheet

	Cost of the product	**Time of usage / devaluation**	**Real cost**
Fixed Costs			
Laptop	*699.99*	*5 months / 36months*	*97.08 €*
Router WiFi	*47.37*	*5 months / 6 months*	*39.48 €*
22' Monitor	*98.87*	*5 months / 6 months*	*82.39 €*
Counselors Salary			*2203.20€*
Junior Engineer Salary			*5040.00 €*
Rental of AWS EC2 Console	*0.00*	5 months / 5 months	*0.00 €*
		Total	*7462.15 €*
Variable Costs			*1492.43 €(20%)*

TOTAL COST: 8954.58 €

TOTAL + TAXES (18%) : 10566.40 €

Table 7.2: Table 5.3: Total costs associated to the project, including with and without taxes

The **time of usage / devaluation** represents the percentage of cost out of the cost of the product concerning to this project

7.2 Project management plan

In order to fulfill the goals expected for this project, as well as prepare the appropriate documentation, presentation and defence, an organized plan was developed at the beginning. However, times predefined for the research and development were not always enough, and some other task not included at the beginning became part of the project itself. As a summary of the whole process, these are the steps followed during the implementation of this project:

- **Learning about the FIWARE IoT Platform**. This task started in January 2015. Learning about the components and start using them with successful results takes a few months. In this time, the idea of the project came out after several reunions between the developer of this project, Andrea Gonzalez Mallo, and the counselor Marcos Reyes Ureña.
  - Period: March 2015, 90h.
  - Total time of meetings, 3h.
- **Exposing the project as an university project**: Once the main idea had been fixed, a reunion with the counselor Victor Pedro Gil Jiménez launched this project as a Bachelor Thesis project.
  - Period: March 2015, 2h.
  - Total time of meetings, 2h.
- **Research period**: After concluding the framework of the project, several coordination meetings with brainstorming ideas between the developer and the counselor Marcos Reyes Ureña, to find an accurate case use to illustrate the project.
  - Period: June 2015, personal research time around 30h.
  - Total time of meetings, 3h.
- **Design of the solution**: Since the idea was already clear, in summertime the development started taking shape. However, during the implementation of the solution, the design was adjusted to the results and issues found, taking more time than expected.
  - July 2015, 8h session.
  - August 2015 - September 2015, four meetings of 1h each.

  - August 2015 - September 2015, changes after meetings, 3h.

- **Development of the solution**: Once the design was clear, the development of the solution took place in several stages as well.
  - Seeking data and choosing the best data sets available, 5h
  - Comprehending those data sets, compressed in CSV, JSON, and SHAPEFILE formats, 20h
  - Preparing sensor simulators for each data set, after comprehending the needs of each service, 80h
  - Developing a gateway backend to connect those sensors with the platform, 20h
  - Developing a plugin frontend to visualize measures in a dashboard, 45h

  A final presentation of the results to the counselor Marcos Reyes Ureña made clear the importance of remodeling several concepts of the sensor simulators developed, which meant a longer development time.

  - Total time of meetings, 2h.
  - Performing the modifications suggested, 40h

- **Catch up reunion at university**: A meeting with the counselor Victor Pedro Gil Jiménez, to catch up about the status of the project and set up the basis of the final essay attached to the project, took place in early September 2015.
  - Total time of meetings, 1h30.

- **Documentation**: Paperwork tasks started in September 2015, when the development was almost ready. When the essay was halfway done, a meeting with Marcos Reyes Ureña was organized to expose the contents of the documentation, where minor changes were suggested.
  - Total time of meetings, 1h.
  - Performing the modifications suggested, 2h.

  The structure followed during the documentation part attempts to show what is the project about, and why Smart Cities was chosen as the main topic. It was written in phases, enabling a full vision of the document from the beginning. The order of the final document though, is different to the order in which it was written, and this is the real time line followed:

- Research about current issues related to the IoT development, to expose the framework that holds the project. Part of the initial scope exposed was made after this research.
  * Time: 9h.
- Research about major vendors of IoT environments, mainly focused in home automation solutions nowadays. Contribution of the expert in the field Carlos Ralli Ucendo, IoT Architect in Telefonica I+D.
  * Time: 8h.
- Research about Smart Cities, their impact and cutting edge examples. This partially helped to the economical and social studio, and set the basis to develop the application designed for the project.
  * Time: 9h.
- Research and documentation of the FIWARE platform, as one of the central pieces of the essay. Reference documentation was the FIWARE whitepaper.
  * Time: 13h.
- Documentation about the development of the project. Goals and motivation was written at the same time, to orchestrate both ideas.
  * Time: 25h
- Final chapters. Conclusions, budget and management, and abstract of this project was written at the end.
  * Time: 6h

To expose the stages while during the whole development, a Gantt charts will be plotted to illustrate the start and finish dates of the terminal elements and summary elements of the project.

7.2.1 Gantt charts

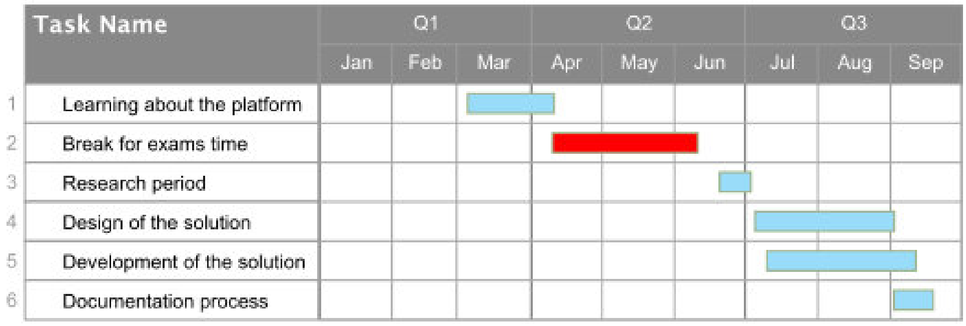

Figure 7.1: Gantt chart representing the whole process, indexed by months.

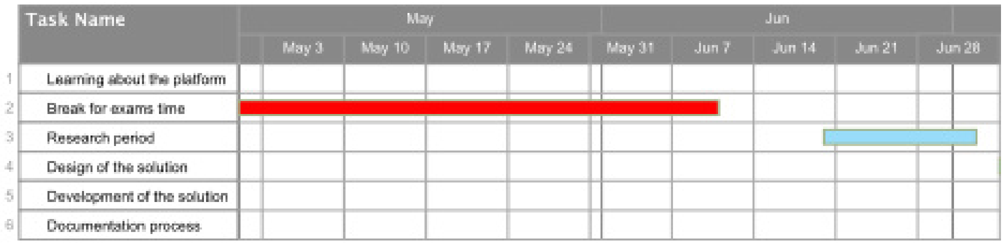

Figure 7.2: Gantt chart representing the intense period, indexed by days. Part 1.

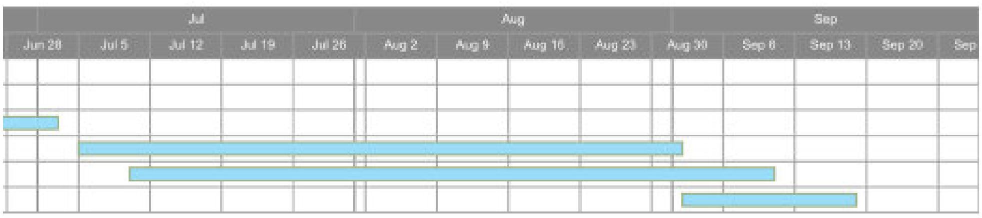

Figure 7.3: Gantt chart representing the intense period, indexed by days. Part 2.

Appendix A

Portal of the IoT Platform

In this appendix, a the total number of options available in the *Portal* website for the IoT Platform are included, with examples of the devices and services created for this project. In order to use this component, a FIWARE account and API credentials to the platform need to be provided. The request to access the *portal* is as follows.

GET `https://<INSTANCE_HOST>:8008`

Once the site has been accessed, a login page will ask for three input parameters: **Username**, **Password** and **User Domain**.

Inside the portal, there are two separate management interfaces: on the one hand, a global management part to handle subservices and users. On the other hand, a management interface for each of the subservices available. A small menu on the right side of the webpage allows to jump from one to another.

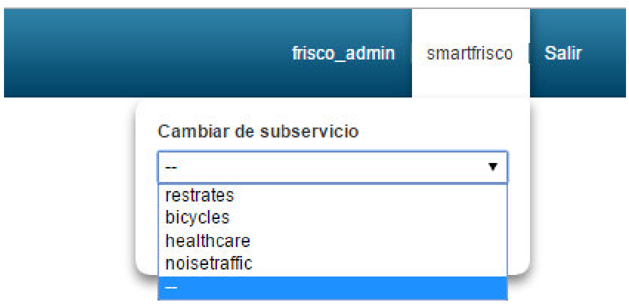

Figure A.1: Global management or Subservice management menu

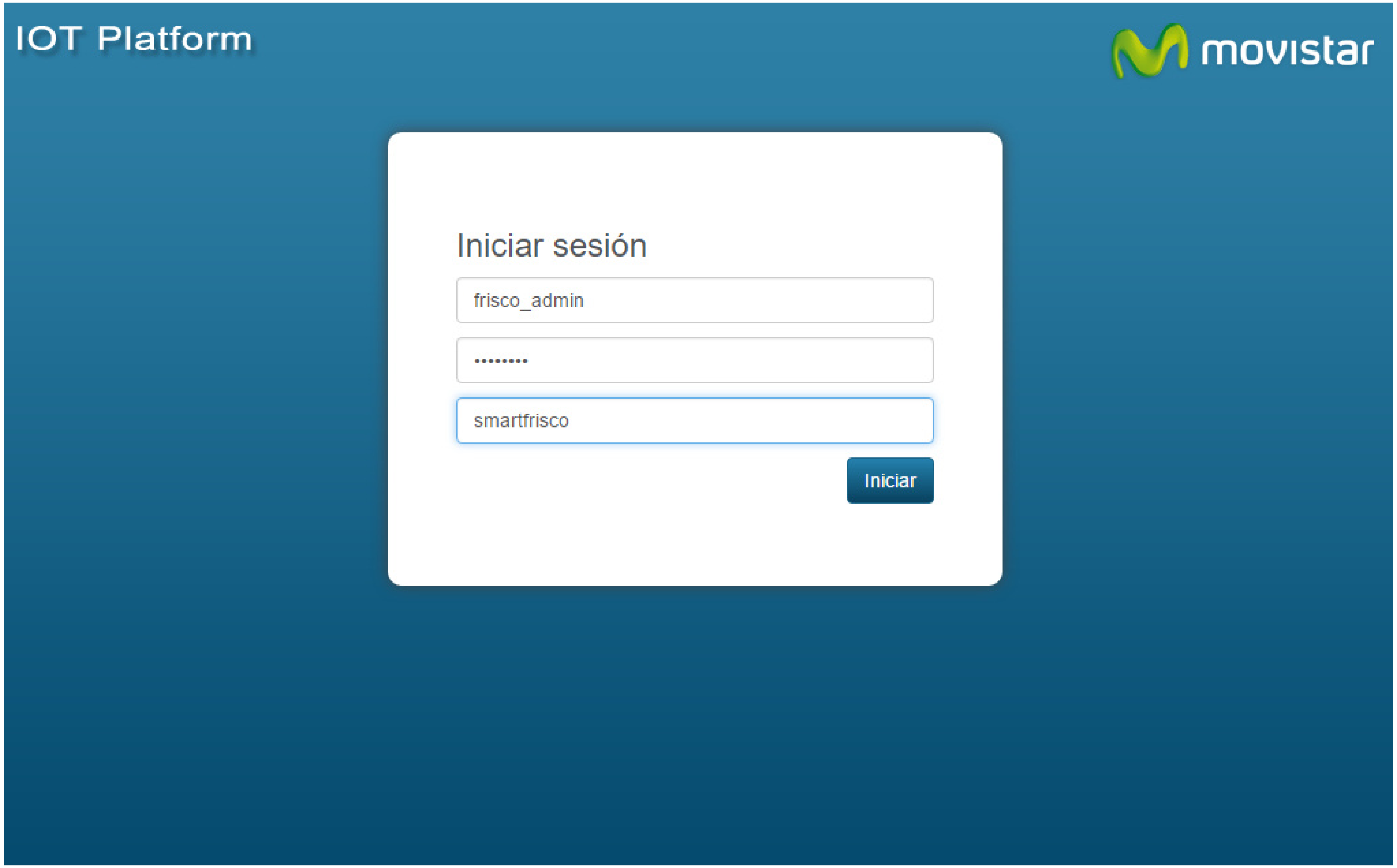

Figure A.2: Login widget to access the portal

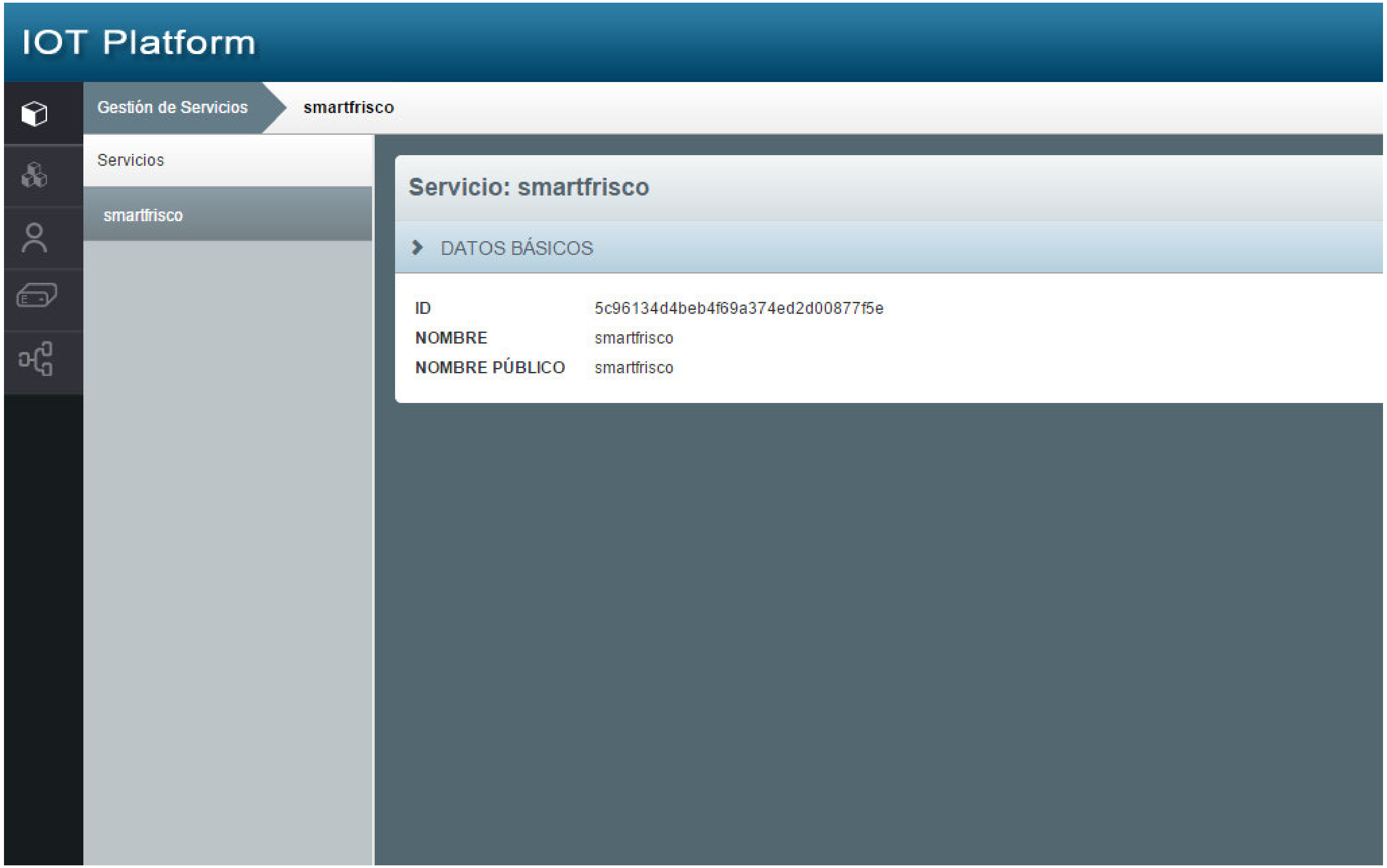

Figure A.3: Service management, only for visualizing

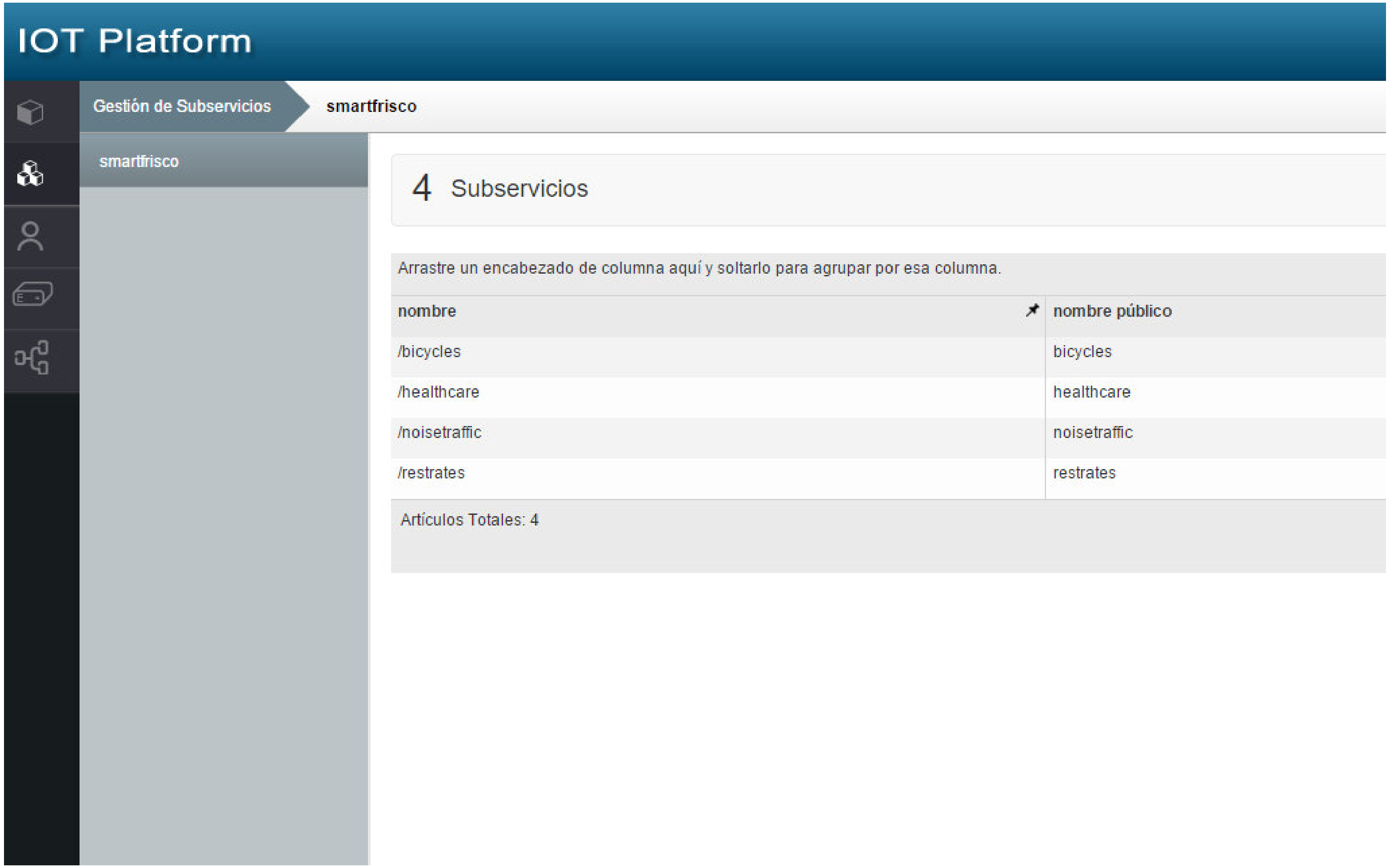

Figure A.4: Subservice management, allows deleting or adding subservices

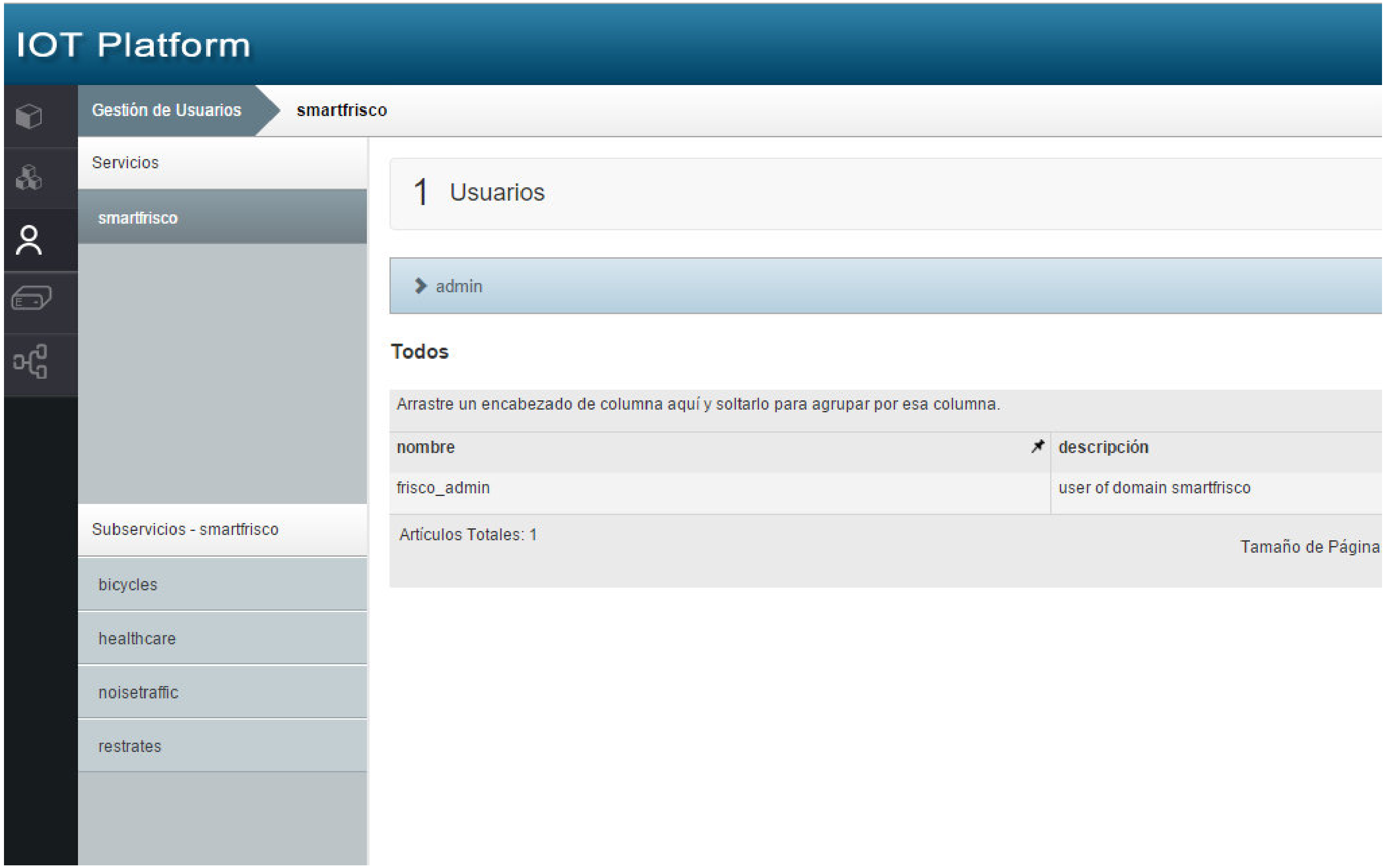

Figure A.5: User management, only add/delete consumer role users

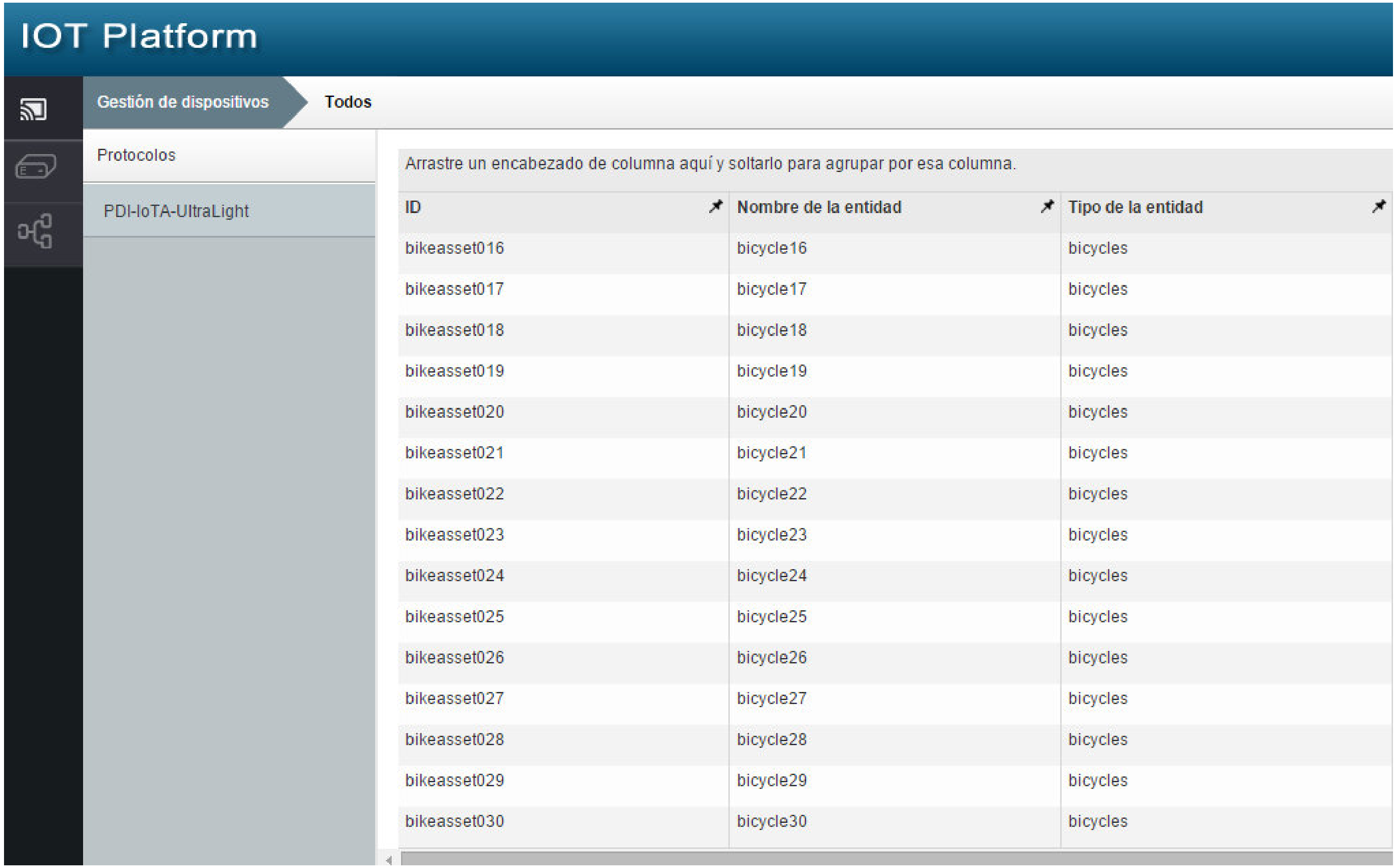

Figure A.6: Device management, inside a certain subservice

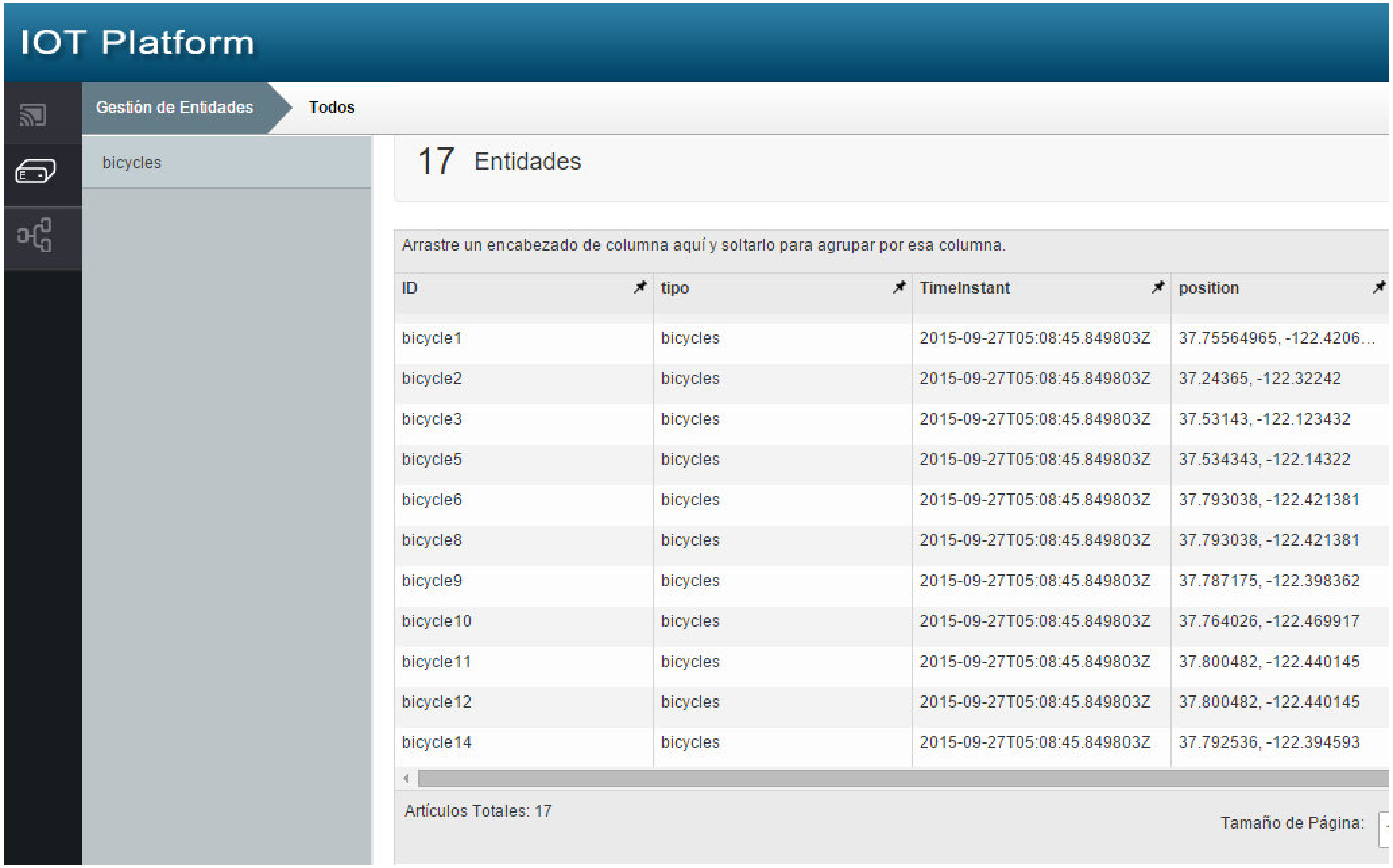

Figure A.7: Entity management, inside a certain subservice

Figure A.8: Rule management example, connection with CEP

Appendix B

Dashboards

One of the goals of this project is the graphical visualization of the devices published in the platform. To do this, a series of dashboards were posted in a personal freeboard account registered with this purpose. The following pages expose the final results.

- **Bicycles dashboard.**

  It contains a map with the bicycles moving around the city in a certain time, and a graph indicating the percentage of bikes in movement in the captured moment.

- **Health centers dashboard.**

  It contains a map with the public health centers in the city, and each of them expose useful information such as name of the health center, rate by previous users, and the number of available appointment slots for the day.

- **Noise traffic dashboard.**

  It contains a map with the average noise captured in an hour in different areas of the city, and also a small capture of the maximum noise peaks in all the areas of the city.

- **Restaurant inspections dashboard.** It contains a map with the business inspected, marked in colors for each of the rates given. It also presents an averaged record of the daily rates.

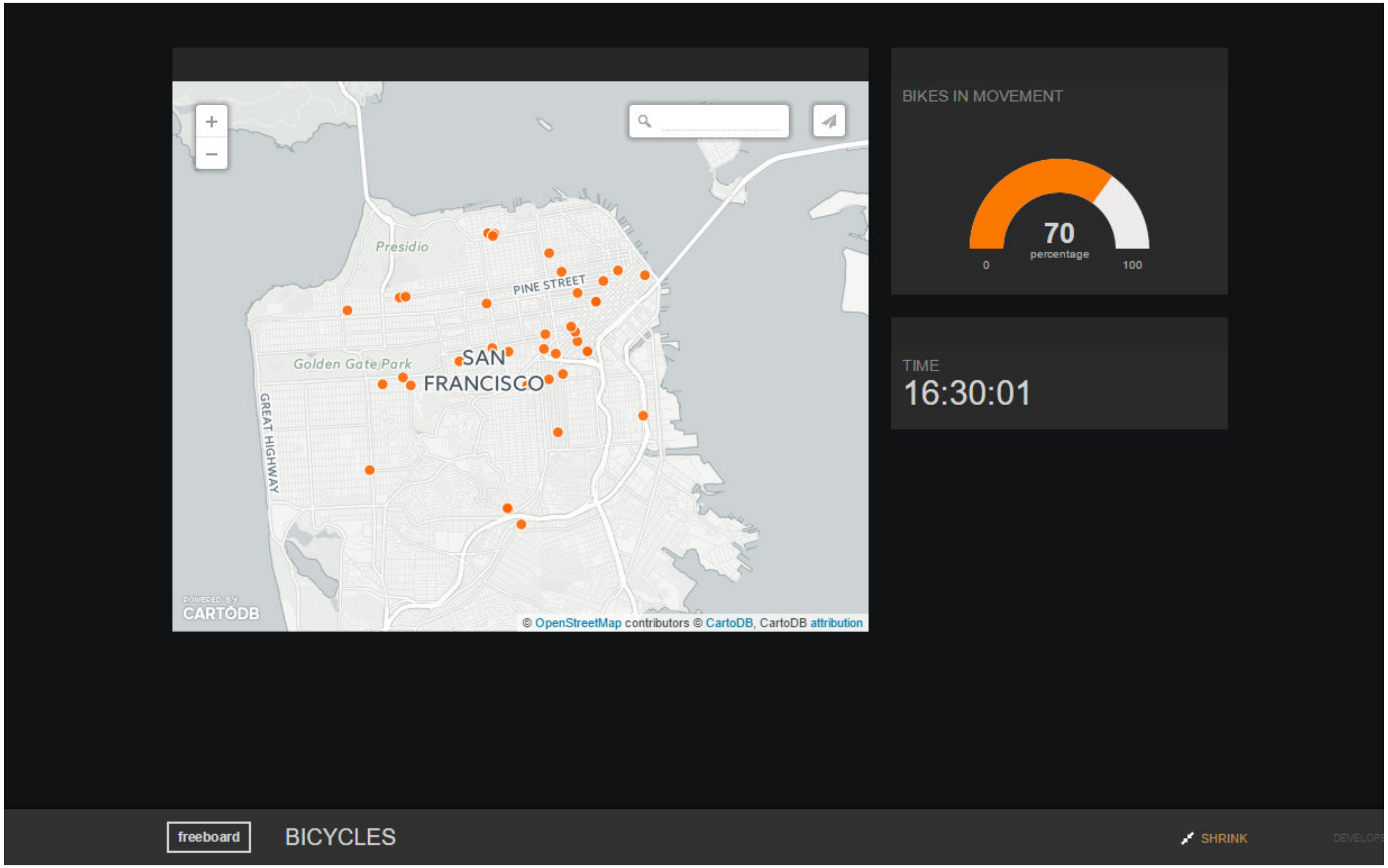

Figure B.1: Dashboard containing the public bicycles in the city

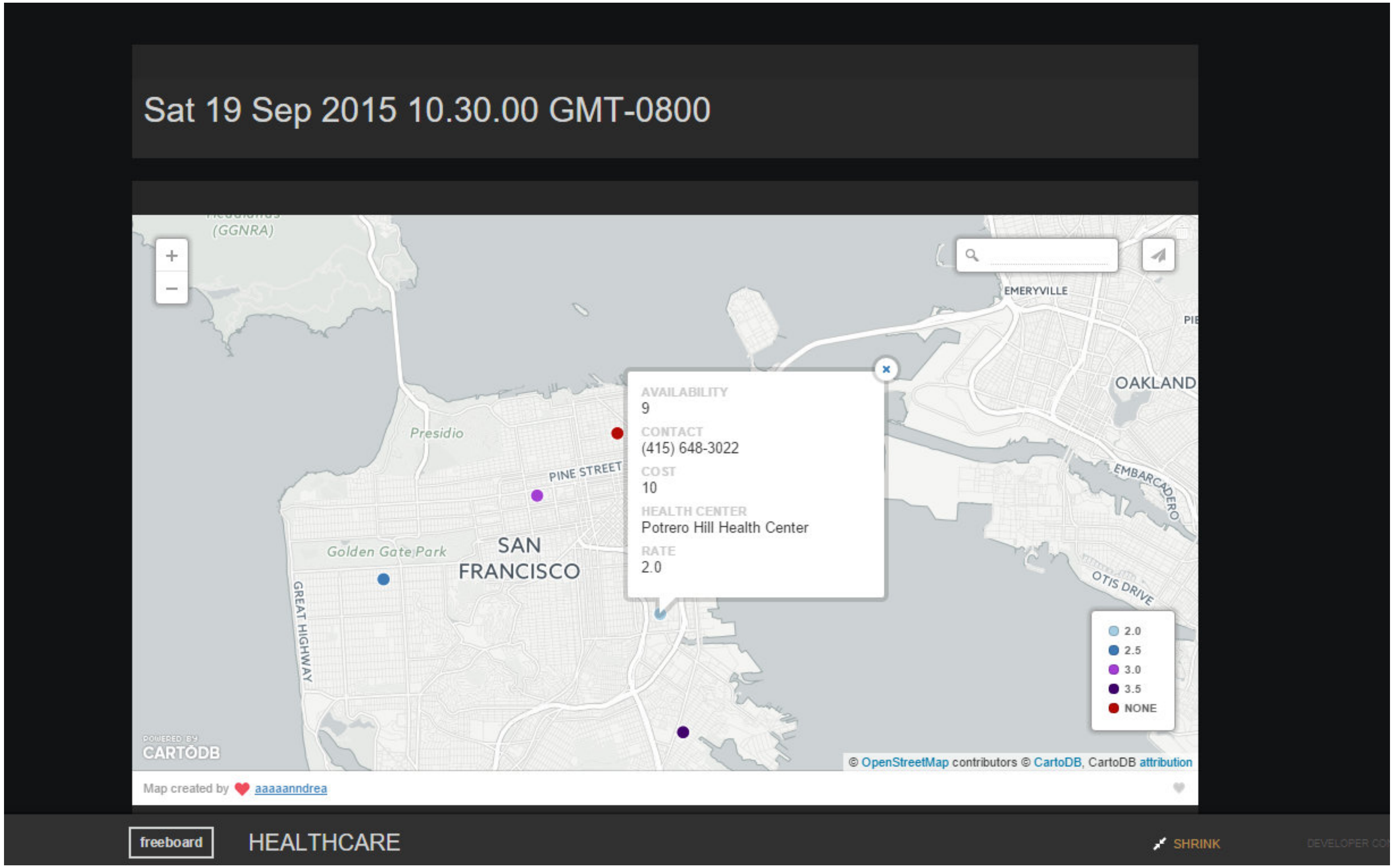

Figure B.2: Dashboard containing the public health centers with flu vaccination service in the city

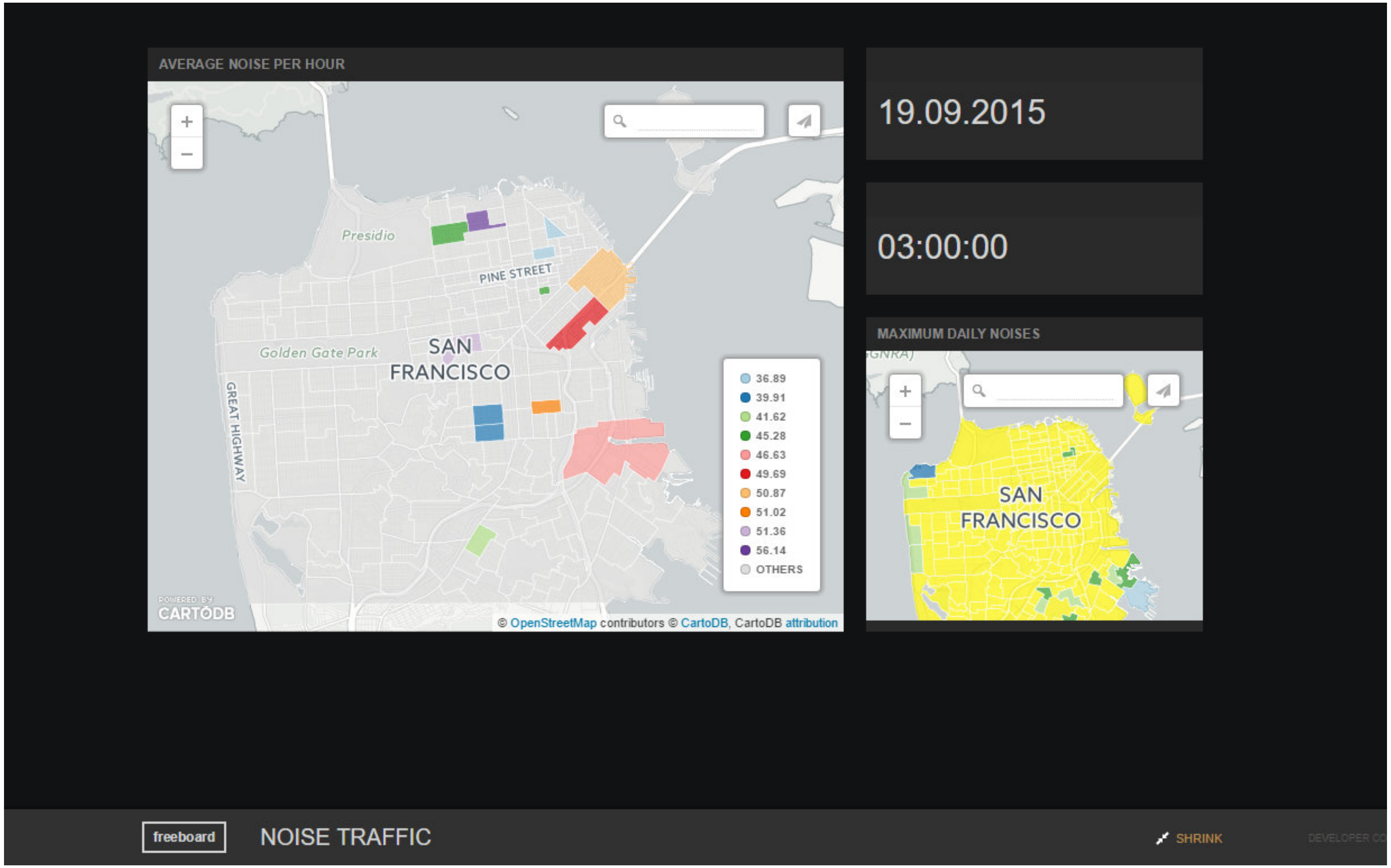

Figure B.3: Dashboard containing the traffic noise measured in decibels in the city

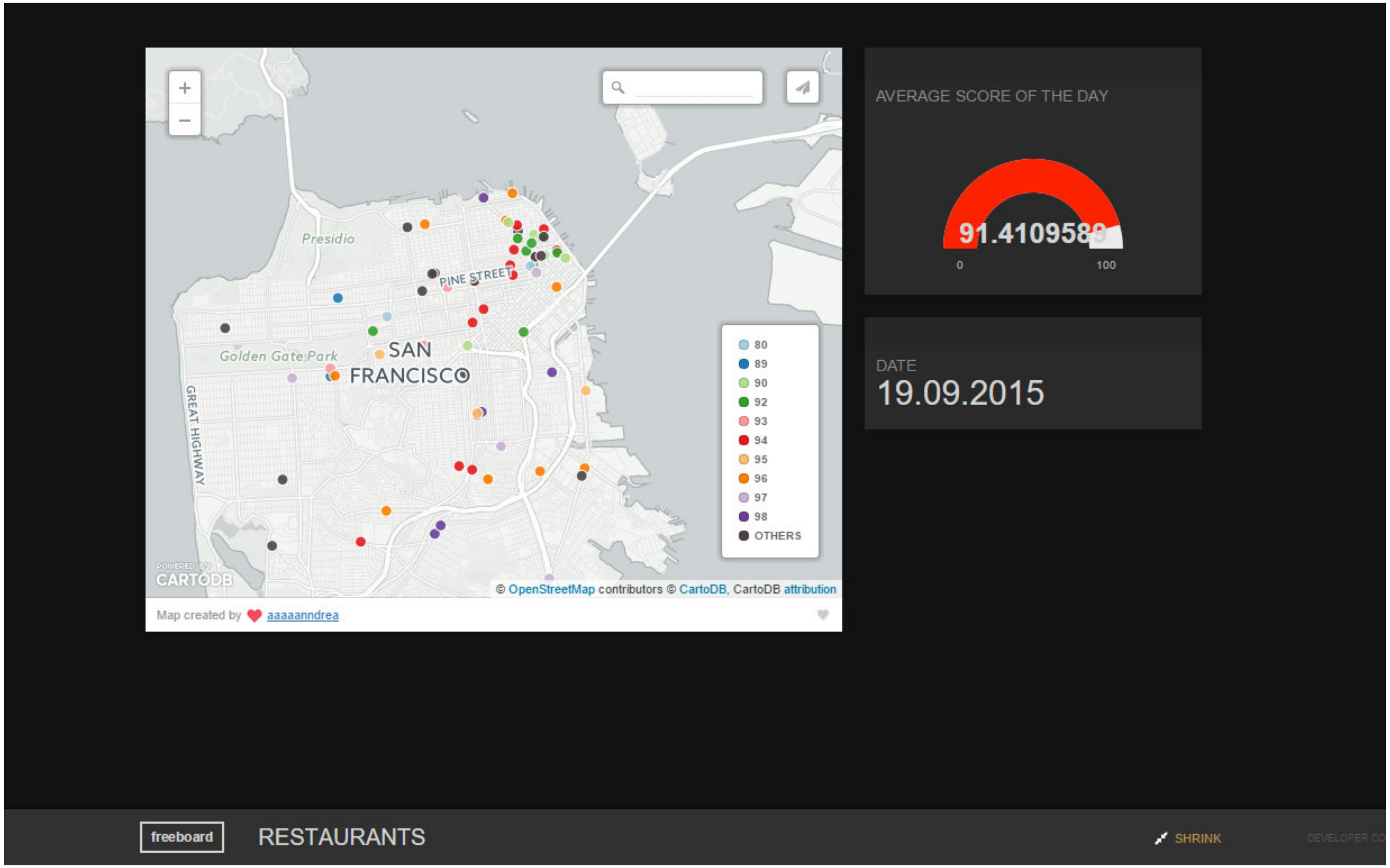

Figure B.4: Dashboard containing the restaurants daily evaluated by inspectors

Bibliography

[1] http://www.elmundo.es/economia/2015/03/29/55172431e27 04e132a8b456f.html

[2] https://www.smartgrid.gov/the_smart_grid/

[3] http://ethic.es/2013/02/construyendociudadesinteligentes/

[4] http://thinkprogress.org/climate/2015/09/08/3699273/citiessaving-moneybyfightingclimatechange/

[5] http://www.rcysostenibilidad.telefonica.com/blog/2014/04/07/ayudanos-adefinirsmartcitysostenible/#.VfRFjfntlBd

[6] http://www.digitalmeetsculture.net/article/smartcitycreative-sustainableandusercentredcities/

[7] http://www.iec.ch/whitepaper/pdf/iecWPsmartcitiesLRen.pdf

[8] http://www.itu.int/en/ITUT/focusgroups/ssc/Pages/default.aspx

[9] http://www.zabala.es/wpcontent/uploads/2015/docs/2015_02_23_Diario _de_Navarra_smart_cities.pdf

[10] http://www.eldiario.es/hojaderouter/tecnologia/smart-city-espana-negocio-economia-ayuntamientos-empresas_0_355915190.html

[11] http://www.retema.es/actualidad/une-178303-y-une-iso-37120-las-dos-nuevas-normas-sobre-ciudades-inteligentes-de-aenor

[12] http://www.iso.org/iso/smart_cities_report-jtc1.pdf

[13] http://www.iso.org/iso/home/standards_development/

[14] https://ec.europa.eu/digital-agenda/en/our-members

[15] https://www.iotivity.org/downloads/iotivity-0.9.2

[16] http://openinterconnect.org/faq/

[17] http://www.mbed.com/en/about-mbed/what-mbed/

[18] https://developer.mbed.org/blog/entry/Announcing-our-plans-for-mbed-v30/

[19] http://www.industrialinternetconsortium.org/about-us.htm

[20] https://en.wikipedia.org/wiki/Thread_(network_protocol)

[21] http://threadgroup.org/Technology.aspx

[22] http://threadgroup.org/About.aspx

[23] https://developer.android.com/training/building-wearables.html

[24] http://elekslabs.com/2015/03/apple-watch-vs-android-wear-time-to-drive-tesla-further.html

[25] https://developer.apple.com/library/ios/documentation/HomeKit/

[26] https://developer.apple.com/homekit/

[27] https://developer.apple.com/videos/wwdc/2014/

[28] http://www.designnews.com/author.asp?section_id=1386

[29] http://www.androidauthority.com/mbed-client-605295/

[30] http://elekslabs.com/wpcontent/uploads/

[31] http://www.juniperresearch.com/press/press-releases/barcelona-named-global-smart-city-2015

[32] http://www.forbes.com/sites/peterhigh/2015/03/09/the-top-five-smart-cities-in-the-world/

[33] http://smartcity.bcn.cat

[34] http://81.47.175.201/project-protocol/index.php/forum-22-district-heating-cooling-network-2

[35] http://ibarcelona.bcn.cat/sites/default/files/city_os_-_inside.pdf

[36] http://www.cicbcn.cat/

[37] http://opendata.cloudbcn.cat/MULTI/en/catalog

[38] http://www1.nyc.gov/site/forward/innovations/smartnyc.page

[39] http://www.fiware.org

[40] https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/

[41] http://searchsecurity.techtarget.com/definition/single-sign-on

[42] https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[43] http://blog.empowerid.com/blog-1/bid/164625/What-is-federation-And-how-is-it-different-from-SSO

[44] https://github.com/telefonicaid

[45] http://docs.piotp.apiary.io/#orchestrator

[46] http://docs.mongodb.org/v2.4/core/introduction/

[47] http://json.org/

[48] https://hadoop.apache.org/

[49] http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[50] https://data.sfgov.org/

[51] https://freeboard.io

[52] http://www.michaelpage.es/productsApp_es/comunicacion/est udios/estudioremuneracioningenieros.pdf

[53] Nielsen, P. S., Ben Amer, S., Halsnæs, K. (2013): 'Definition of Smart Energy City and State of the Art of 6 Transform Cities using Key Performance Indicators', EU-FP7 TRANSFORM project Work Package 1, Deliverable 1.2, http://orbit.dtu.dk/ws/files/60555188/ Definiton_of_smart_ city_D1.2_FINAL.pdf

[54] http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp