Deployment of an open sensorized platform in a smart city context

Sergio Trilles[a], Andrea Calia[b], Óscar Belmonte[a], Joaquín Torres-Sospedra[a], Raúl Montoliu[a], Joaquín Huerta[a]

[a]*Institute of New Imaging Technologies, Universitat Jaume I (Spain)*
[b]*Department BE-OP, European Organization for Nuclear Research, CERN Cedex (France)*

Abstract

The race to achieve smart cities is producing a continuous effort to adapt new developments and knowledge, for administrations and citizens. Information and Communications Technology are called on to be one of the key players to get these cities to use smart devices and sensors (Internet of Things) to know at every moment what is happening within the city, in order to make decisions that will improve the management of resources.

The proliferation of these "smart things" is producing significant deployment of networks in the city context. Most of these devices are proprietary solutions, which do not offer free access to the data they provide. Therefore, this prevents the interoperability and compatibility of these solutions in the current smart city developments

This paper presents how to embed an open sensorized platform for both hardware and software in the context of a smart city, more specifically in a

Email addresses: strilles@uji.es (Sergio Trilles), andrea.calia@cern.ch (Andrea Calia), belfern@uji.es (Óscar Belmonte), jtorres@uji.es (Joaquín Torres-Sospedra), montoliu@uji.es (Raúl Montoliu), huerta@uji.es (Joaquín Huerta)

university campus. For this integration, GIScience comes into play, where it offers different open standards that allow full control over "smart things" as an agile and interoperable way to achieve this. To test our system, we have deployed a network of different sensorized platforms inside the university campus, in order to monitor environmental phenomena.
Keywords: Internet of Things, Smart Cities, Open-hardware, GIScience, Environmental monitoring, GEOEvent

---

1. Introduction

The current century is perceived to be the century of cities. Some studies indicate that in 2007 the urban population exceeded the rural population in the world, and forecasts indicate that in 2050 it will be nearly 70% [1] of the total population. Cities have a huge impact on the economic and social development of nations. They are environments where people live and work, where companies carry out their activities, and are also major centres of resource consumption. It is estimated that nowadays cities are responsible for consuming 75% of the world's energy and generating 80% of the gases responsible for the greenhouse effect [2].

In an urban environment with a growing demand for efficiency, sustainable development, Quality of Life (QoF) and an effective management of resources, public authorities must consider an evolution in the way that the cities are managed; so, new ways of managing are necessary to optimize resources. These are the objectives of a smart city. The use of Information and Communications Technology (ICT) is essential to lead the smart city movement [3]. Moreover, the latter requires advancements to connect everything (sensors, actuators, etc.) as well as to monitor and act [4]; this process has come to be called the Internet of Things (IoT).

Smart city is defined as the city that uses ICT to make its infrastructure, components and utilities offered more interactively and efficiently so that citizens can become aware of them. ICT has to play a crucial role in designing smart city projects and will be the key for supporting the transition to more sustainable cities [5]. Smart cities are defined by six different factors, such as: economy, people, governance, mobility, environment and living [6]. All these factors must be taken into account when we are considering deploying a smart city. IoT describes how in the real world, physical things are integrated into the digital world of bits and bytes [7]. In general, it involves the integration of each object, such as a sensor or device, that is connected via wired or wireless networks using the Internet. To achieve connectivity between things, the Internet Protocol (IP) is used in which each device has its own IP address.

In a technological context, the smart city and the IoT concepts are two terms that are very closely linked. Both concepts need a mechanism to communicate; it is called Machine to Machine (M2M) and it will work with new developments and contribute to the future of Internet [8]. Actually, the future of the Internet will consist not only in connecting people, but also on the approach of a digital world, in which, ideally, everything will be connected. In the context of a city, these things can be buildings, cars, appliances, meters, etc. Undoubtedly, this new “real network” will lead to a new way of managing a household, company, community, city or even country.

Smart cities need IoT to listen and to understand what is going on in town, thus, allowing for better decisions and providing information and services appropriate to their inhabitants. In addition, the use of advanced real-time analytic techniques is what creates a sort of awareness and understanding of the city, which undoubtedly improves services. Current smart cities are urban spaces with infrastructure, intelligent networks and platforms, with millions of sensors and actuators, in which the people themselves and their mobile phones must also be included.

For all of these reasons, there is an increasing demand for the deployment of Wireless Sensor Networks (WSNs) that provide updated information about the state of the environment not only in a city, but also in many other scenarios, like agriculture, meteorology and/or health.

Nowadays, there are initiatives to monitor cities (or other scenarios), which are based on commercial solutions [9] and are usually expensive and proprietary; this can involve some barriers and preclude their continued development. Faced with these impediments, there have been many initiatives of open embedded platforms which have become important [10]. In recent years, there have been several projects that can connect with Internet such as "smart thing" and they have released the schematics of their devices, which has increased their use. One remarkable example is the Arduino project [11], a low-cost and easy to use micro-controller platform, with a huge community of developers that share information, experiences and knowledge.

Within this context, we present our proposed method, which consists of the deployment of an open (both hardware and software) sensorized platform called *SEnviro* [12] in a smart city context. To achieve this, we have used *SmartUJI* [13] as a test-bed to check the possibility of adding an open platform in order to meet the requirements of obtaining a smart city. *SmartUJI* is an initiative to apply smart cities' approaches in a university campus (Jaume I University's[1]). It is based on the integration of the university's-corporate information, digital maps and derivative applications.

In this work, the union between IoT and smart cities will be addressed in the hands of Geospatial Information Systems (GIS) and GIScience. It will give us the methodologies and tools to make this union possible. GIScience is called on to support the development of the "smart devices" in order to enhance its compatibility and interoperability.

A new solution is used to make the said union possible. As we will detail in the next sections, an extension of ArcGIS Server[2] is used which is called GEOEvent[3]. This tool can be used as a bridge between the two concepts. Also, it adds new improvements such as enabling data consumption in real-time and applying algorithms over these data in order to launch alerts or generate some outputs as a result.

In summary, the main contributions of this work are: (1) integration of an open hardware sensorized platform in a smart city context; (2) GIScience standards and solutions have been applied to enhance interoperability between IoT and smart city solutions; (3) applying for monitoring environmental meteorological and air quality phenomena; and (4) offering a client in order to make decisions over data provided by sensors.

The rest of the paper is organized as follows. Section 2 presents the

[1]See http://www.uji.es/ (last accessed: November, 2015)

[2]See http://server.arcgis.com/ (last accessed: November, 2015)

[3]See http://server.arcgis.com/en/geoevent-extension/ (last accessed: November, 2015)

background to position the current work. Section 3 details the smart city platform, called *SmartUJI*. Section 4 talks about the *SEnviro* platform, an open sensorized platform to make "smart things". Section 5 shows how we have integrated the *SEnviro* platform in the *SmartUJI*. Section 2.5 enumerates similar related work. The paper ends in Section 6 with conclusions and future work.

2. Background

This section details the concepts that will be employed throughout the article. It talks about smart cities and the different factors that are featured. Afterwards, the IoT movement using open-hardware is discussed. Subsequently, the third section details how the cloud computing concept is applied to the IoT environment. Finally, the last section details how GIScience is able to standardize the IoT movement.

2.1. Building smart cities

Although there is still no formal definition and widely accepted "Smart City", the ultimate goal is to make better use of public resources, increasing the quality of services offered to citizens, while reducing operating costs within the public administrations. This can be achieved by deploying a network of IoT devices.

The IoT movement aims to make the Internet even more enveloping and pervasive in our daily lives. It allows easy access and interaction with a wide variety of devices such as, for example, appliances, cameras, sensors, actuators, displays, vehicles, among others.

Actually, this paradigm is applied in many different fields, such as home automation, industrial automation, medical aids, mobile healthcare, elderly care, intelligent energy management and smart grids, automotive, management fields traffic, and many others [14].

Nonetheless, it is a very heterogeneous field, because there are solutions that meet the demands of all possible scenarios. This situation has led to the proliferation of different and sometimes incompatible proposals for the realization of a common goal; as a result the search for a common standard is one of the biggest challenges.

The availability of different types of data collected by a network of IoT devices can also be exploited to increase transparency and promote the local governments interaction withits citizens, increase public awareness of the situation of the city, encouraging active participation of citizens in managing public administration, and to stimulate the creation of new services.

In addition to increasing citizen participation, the main benefit of this move is to add value to existing services [15]. The challenge is to offer services within cities to increase quality and improve life, and this implies an economic advantage for administrations. Examples where the IoT devices are applied and provide benefits to both citizens and administrations are: structural health of buildings, waste management, air quality monitoring, noise, traffic congestion, energy consumption, smart parking or smart lighting.

These examples are part of some of the six axes of a smart city [4]. These axes are: urban mobility (smart mobility), energy efficiency and overall sus-

[4] http://www.smart-cities.eu/ (last accessed: October, 2016)

tainable resource management (smart economy), participatory governance (smart governance), quality of life (smart people), attractive natural conditions, pollution or environmental protection (smart environment), public safety as well as the areas of health, education and culture (smart living). Subsequently, each of these factors are detailed.

According to the study by [16], smart city initiatives related to the smart environment (33%) are those that have more representation, followed by those that are associated to smart mobility (21%). The other factors are distributed more evenly, registering a percentage of about 10% coverage in all cities. These numbers reveal that there is a great concern about environmental monitoring [17, 18, 19, 20]. In fact, there are multiple projects that have been developed to quantify or qualify the environment to address this concern and they use sensors to collect observations from the environment.

2.2. Open-hardware to make smart things

On one hand, the main ICT providers have made efforts to develop cross-platform intelligent cities (PlanIT operating system[5], the IBM Intelligent Operations Center[6] or Smart City platform solution Oracle[7]) designed to integrate information from the city and make it available to end users.

On the other hand, utilities (water, electricity, gas, etc.) have their proprietary solutions (SCADA, Amis, etc.) specifically designed to operate and

[5] See http://living-planit.com/ (last accessed: October, 2016)

[6] See http://www-03.ibm.com/software/products/en/intelligent-operations-center (last accessed: November, 2015)

[7] See http://www.oracle.com/us/industries/public-sector/national-local-government/city-platform/index.html (last accessed: October, 2016)

monitor this infrastructure and provide management services and billing. This situation poses a problem of interoperability between the mainframe and other specific systems. The lack of a common standard and not knowing how both systems are built, preclude their integration.v

As a result of this, nowadays the open movement has gained a lot of attention [10]. Traditionally, this idea was applied more freely to the code and provoked the open-source movement. More recently, the same idea has been applied to the hardware (open-hardware). This situation has permitted offering complete open systems, where both source and hardware are open. The users can contribute to improving the system and propose improvements. One of these improvements is the capacity to offer more interoperability between components, because this approach can be adapted for each usage. So, it reduces compatibility issues. This advantage permits the application of different standards that in the future can be established and will allow a standardized approach to the IoT. In this work, we apply different GIS standards to communicate with each IoT device; we will talk about these standards in the following paragraphs.

Another improvement is the low cost of these devices compared with commercial solutions. As concluded by [21], open initiatives are cheaper, but have also demonstrated that open hardware is more powerful and has better performance than commercial solutions. Also, the same work demonstrates that the open hardware project can be considered reliable.

Currently, there are several different alternatives regarding the micro-controller based platform, and the most remarkable platforms are: Arduino, Raspberry Pi, BeagleBone or MSP430 Launchpad. All these platforms can be easily expanded, and offer other robust applications.

Different applications have been developed thanks to the acceptance of IoT movement. Apart from smart city applications there are others that also contribute to attaining a smart city. These are healthcare [22, 23, 24] (ambient-assisted living and telemedicine), smart home and smart metering [25, 26] (devices that enable the automation of common in-house activities), video surveillance [27] (intelligent video systems), automotive and smart mobility [28, 29] (intelligent transportation systems) and smart energy and smart grid (intelligent management of energy distribution and consumption) [30].

All these applications have produced the popularity of "smart things" connected to Internet, and the networks created with them. In this way, the traditional WSNs have been updated to follow an IP connection in order to apply all the benefits that it entails. The new paradigm, namely cloud computing, has been added to improve and provide more advantages over the IoT [31].

2.3. IoT in the cloud

Smart solutions need the two paradigms, IoT and Cloud Computing (CC). They had followed different paths, but in the last years, IoT has been supported by CC. CC offers a good solution to implement IoT service management [30], and gives unlimited computational and storage capabilities that are limited in IoT. Also, CC can take improvements from IoT, extending its scope to "real things". These two worlds are forced to understand each other.

There are many challenges and opportunities for emerging future smart cities, which can be addressed by means of cloud computing [32, 33]. The IoT movement is distinguished by a wide variety of devices, with different technologies and protocols. These features mean different disadvantages, such as the lack of interoperability, security, scalability, efficiency or reliability. It is difficult to achieve all these characteristics, but the cloud approach may be helpful to solve these shortfalls by offering additional features like ease-of-access and use with lower cost [34, 35].

CC offers great benefits for applications hosted on the web, which also have special computation and storage requirements. In addition, it offers easy access to them via networks and allows users to focus on the development of applications. Users only have to set up their sensors to send data through IoT platforms.

The cloud can provide a virtual infrastructure that integrates device control, storage, analysis tools, visualization, and delivery of customer platforms. This platform acts as a receiver of ubiquitous sensor data, to analyze and interpret the data, as well as to provide the user with a visualization that is easy to understand, which is a key feature required for the generation of IoT applications based on the web.

Figure 1 details the work-flow for connecting sensor nodes to users using a cloud computing infrastructure. This figure divides the physical and cyber worlds and the IoT part with traditional Information Technology (IT). In order to pass the barrier between the IoT device cloud and the IT cloud, a gateway is used [36]. There are many options available for the gateway devices. It can be used to perform routine tasks. It has also been used to add security features, like encryption. We use this approach to deploy the sensorized platform detailed in Section 4.

2.4. GIScience role in IoT

The third and last pillar to understand the proposed work is GIScience. GIScience can be defined as "the discipline that uses Geographic Information Systems as tools to understand the world" [37]. It can contribute by offering an important key component for the IoT movement, which is interoperability by means of the use of standards. A non-profit international organization which develops open standards for the global geospatial community is the Open Geospatial Consortium (OGC)[8]. OGC defines standards in a wide variety of domains including environment, defence, health, agriculture, meteorology, sustainable development, and smart cities.

The OGC has standardized Sensor Web Enablement (SWE) as a set of specifications related to sensors, sensor data models and sensor web services that will enable sensors to be accessible and controllable via the web [38]. It is composed of a set of standards, the most popular are: Observations and Measurements (O&M), SensorML, Sensor Observation Service (SOS), Transducer Model Language (TransducerML), Sensor Planning Service (SPS), Sensor Event Service (SES) or Web Notification Service (WNS).

Other proposal is OGC SensorThings API [9]. It is a new candidate standard that provides open access built on web protocols, based on the current SWE standards and following the architectural REST style. Its aim is to provide a standardized method to expose the world of the IoT to the real

---

[8]See http://www.opengeospatial.org/ (last accessed: November, 2015)

[9]See http://ogc-iot.github.io/ogc-iot-api/ (last accessed: November, 2015)

world, where things have limited resources.

All these standards have been developed following the open software initiative. Matter of fact, they are the different available implementations. The highest representative of them, is 52 North [10]. It supports a lot of the standards, and offers them as open software.

If we consider proprietary solutions related with GIScience, the main player is ESRI Inc[11]. This corporation offers a wide variety of products. All these products are fully compliant with OGC specifications (WMS, WCS, and WFS) in addition to other industry standards. The ArcGIS Server is the heart of all of these products and is used to publish and share geospatial data, maps or models, among others. Afterwards we detail how this technology is used to deploy *SmartUJI* and to connect a IoT solution with a smart city platform.

GIScience can offer a good opportunity to standardize the ways of connecting to the different "things", offering an interoperable connection; and it should help to deploy the new smart cities.

2.5. Related work

In the literature, there are some approaches that are similar to our proposal. The following items describe the works that have been analyzed. All of them talk about IoT in the smart city context.

- [39] describes the SmartSantander. The objective is meant to measure different environmental parameters like temperature, CO, or light and

[10]See http://52north.org/ (last accessed: June, 2016)

[11]See http://www.esri.com/ (last accessed: November, 2015)

present this information within a web interface.

- The authors in [40] simulate a real-world scenario to test a framework to connect IoT devices to the cloud. This framework is called Stack4Things and uses the MOM concept to serve the data provided by the simulated sensor (Ceilometer).

- In [41] the authors present an implemented data acquisition system based on an IoT framework.

- The work presented in [42] presents the design considerations, a real implementation and validation of a system called smart signal. They use an EEPROM memory with Zigbee communication to implement a system to control the emergency signaling in a tunnel context.

- A cloud based general architecture for smart cities is presented in [43]. It tests the platform with a cloud based service for on-line health analytics for use within the community to support personalised on-line health analytics.

- [44] describe a pilot implementation of urban IoT within a smart city framework in Podava. They propose an IoT peripheral system which supports an HTTP-based interface, making it possible to interact with it in an open, standard, and technology-independent way. Each IoT provides different phenomena, such as light, humidity and benzene level in the air.

- Pokric et al. [45] present an environmental monitoring solution called ekoNET to monitor the air pollution and atmospheric parameters. The authors declare the solution as end-to-end and it offers all necessary components, namely devices, back-end infrastructure and client applications both web and mobile.

In order to compare the formerly reviewed works, Table 1 has a comparison between the detailed works. The following features to characterize each one have been proposed:

Work Reference	Open hardware	Open software	Comm. technology	Comm. protocol and format	Real-time	GIScience tools	Smart factor
[39]	Waspmote	✗	IEEE 802.15.4	N/A N/A	✗	✗	Environ.
[40]	Arduino[12]	✓	N/A	Pub&Subs Raw data	✓	✗	Environ.
[41]	RaspPI	✓	Serial	N/A Raw data	✗	✗	Environ.
[42]	EEPROM	✓	Zigbee	Polling N/A	✗	✗	Economy
[43]	RaspPI	N/A	Bluetooth	Polling N/A	✗	✗	People
[44]	N/A	N/A	6LowPAN	Polling Raw data	✗	✗	Environ.

Table 1: Comparison between different related works.

- Open-hardware: the work uses or does not use an open-hardware components to create the “smart things”.

[12] Simulate

- Open-software: the system is offered or is not as open-software. Scale: Yes/No.

- Communication technology: what kind of infrastructures is used to connect the "smart things".

- Communication protocol and format: what kind of protocol and format are used to give connectivity for each "smart thing".

- Real-time: indicates if the system works in real-time. Scale: Yes/No.

- GIScience tools: shows if the system offers GIScience tools as OGC standards, GIS layers, etc.

- Smart factor: smart environment, smart people, smart economy, smart mobility, etc.

Table 1 shows that Arduino and Raspberry Pi are used in three works, this corresponds to half of the analysed works. These two platforms are easy to use because they have large communities that support the two platforms and offer a wide variety of examples and tutorials.

The works proposed in [40, 41, 42] use open-software to deploy their systems. Our project uses partially open-source, because the GEOEvent extension is considered as proprietary software. All other parts included *SEnviro APP*, can be considered as open-source.

The technology used to connect the "smart things" is different for each analysed work and it depends on the communication infrastructures available in the scenario to be deployed. In our case, we used Wi-Fi because, our use case offers Wi-Fi connectivity around the whole area.

Our work follows different communication protocols between them, we offer a Publish & Subscribe with the *StreamLayer*, which offers a real-time connection. However, unlike the others, our project aims to be interoperable, and for this purpose, it is the only one that uses GIScience standards, such as SOS and SensorThings API and we provide different layers such as *FeatureLayer*.

Finally, we can conclude that smart environment is the main smart factor where "smart devices" are used; this is based on the fact the objective of five out of seven analyzed papers has been to monitor environmental parameters.

3. *SmartUJI* platform

The objective of this section is to describe the current *SmartUJI* platform [13, 46]. To test our system, we have used a university campus as smart city test bed. A University campus can be considered a small city as it provides similar services and has similar infrastructures, communications, and transport networks, all of which are necessary for managing and carrying out the daily activities of thousands of people (students, staff and visitors), much in the same way they do in any small city.

Although *SmartUJI* may be wrongly considered as a simple collection of base maps and other kind of data (see Figure 2), it is a platform where all the public information related to the university has been standardized and made accessible using RESTful and web services. Geospatial location has been used to join the campus topology and these other kinds of data. Thereby, we demonstrate this on a map, which shows where each professor is teaching his/her class or by the buildings power consumption

SmartUJI aims to help students, staff and visitors by increasing efficiency. It is intended to serve as a platform for decision-making within the University campus. Another functionality of the *SmartUJI* is to offer access to different datasets, where they present some natures (formats) and codifications. In this way, it offers one access point to multiple kinds of data. To serve and visualize all these data, *SmartUJI* implements different services and offers some applications for each functionality. The *SmartUJI* pretends to be a horizontal/transversal platform with interoperable data and services, in order to offer open access and facilitate the creation of applications.

To build this platform, we have used a *SmartUJI* Core that will be used as a base to expand and create new applications. This Core is built with a Commercial Off-The-Shelf (COTS) GIS software, called ArcGIS Server. It is a powerful tool in geospatial and mapping applications which was developed by Esri Inc. Some benefits that are achieved by offering this core, are: scalability, extensibility, reliability and/or security. Also, we obtain an open, interoperable and standards base system [47].

Following a GIS point of view, we can consider a smart city platform as a Spatial Data Infrastructure (SDI) [48]. In the SDI context, the architecture is defined with three different layers, as they are content, services and applications (Figure 3).

3.1. Content layer

In this first layer, data and models are stored. In a smart city context, the data layer will contain the structure and the relationship of all the entities that are required for this model, in this case, a University campus. Each modelled element was stored in a main geodatabase and has features, such as: area, building identifier, floor, side and room number. These features were used to join geospatial elements and other information. Afterwards, these data were in different sites and formats (txt, csv, pdf, etc.). This platform has only been used to offer an access point.

3.2. Services layer

Contains basic maps and data services. Several services have been developed to retrieve the interior and exterior cartography of the campus. In addition, services have been developed to get information from external databases. Other services are provided in order to facilitate access to the data and to visualize them in layers. Some of them are: services to retrieve information about technical services, academic and research staff from the university directory; a service to provide data about power consumption from smart energy meters; or a service to generate the route between two places. Also, we offer a Wi-Fi Indoor Positioning System (IPS) [49, 46] as a service. All these services can be used by applications, and any changes and improvements are transparent to the mobile application developers and to the final users.

3.3. Application layer

The *SmartUJI* offers a wide variety of applications which have specific functionalities. Among these, a webmap-based called *SmartUJI web* (see Figure 2), that allows query and visualization of the university's database records. It uses all services detailed above. *SmartUJI APP* (see Figure 4 a)): a native smartphone application which provides most of the functionalities of the web-based version. Also, *SmartUJI Augmented Reality (AR)* (see

Figure 4 b)): this application can visualize the location of selected facilities superimposed on the real phone-view as well as guide users through the campus with its route displayed in real time. All applications use the services provided by the Core layer described above.

4. The *SEnviro* platform

This section presents the sensorized platform used to be integrated in a smart cities context. As commented in the above sections, the platform *Sense Our Environment* (*SEnviro*) [12] follows the IoT and Web of Things (WoT) [50] paradigms by means of a low-cost, open-hardware and open-software, energetically autonomous and interoperable solution. Each node of the *SEnviro* platform is called *SEnviro Thing*. The *SEnviro* platform uses the IP protocol to establish the connection. With these features, it can be considered that each *SEnviro Thing* is a smart object. A network is formed by joining several *SEnviro Things*. This platform makes use of standards when publishing data. This guarantees the interoperability of the services provided by the platform.

If we analyse *SEnviro* from a GIS context and specify what is happening in each of its layers, we obtain the Figure 6, which we are going to detail below. In the content layer there are different *SEnviro Things* that cover the area so that the sensors can measure the phenomena in each sensorized platform. When a *SEnviro Thing* gets a new observation, it invokes the Sensor Data Management (SDM) module. This component is implemented using a RestFul web service.

The SDM module is responsible for publishing observations in different services that we want to offer, such as SensorThing API, SOS or others. Thus, it offers the same content using different interface; depending on the context. Thus, different adapters for each of these services are offered.

Thereby, we are able to offer the same content using a different interface and these interfaces will be used depending on the appropriate context. Different adapters for each of these services are offered. If we want to support new services/standards, we should only add a new adapter for the service. We use the adapter software pattern to implement this feature.

The service layer, offers the services described above. One of the objectives of the *SEnviro* platform is to offer a standard service, in order to provide connectivity in an interoperable way. To meet this challenge, a candidate OGC standard, SensorThings API, has been used to offer an interoperable service to access the *SEnviro* platform. The most remarkable thing about this standard is that it breaks away from all "standard topics" [51] and offers access to restrictive devices, such as smartphones. Also, we offer a SOS interface to increase the interoperability, because this standard is widely used [52]. Finally, a Lightweight RestFul service is created to connect this service in a more agile way[53] and will be used to integrate *SEnviro* and *SmartUJI* platforms. The next section will present the different technologies and details.

The last layer, application, is for the different clients, mobile, web-based and/or desktop. As a first proof of concept, we have included environmental sensors [12], such as temperature, humidity and CO2, among others (Figure 7) to the *SEnviro* platform.

5. Integration of *SEnviro* in *SmartUJI*

This section details how we have deployed the sensorized platform in a smart city context. The last two sections have detailed both *SEnviro* and *SmartUJI*, subsequently, we present all steps defined to integrate both platforms. As we commented above, we have built five *SEnviro Things* to monitor environmental phenomena. They are installed around the campus. Each *SEnviro Thing* takes measurements on the different phenomena depending on the sensors installed in it. As detailed in Section 3, *SmartUJI* uses ESRI software for its deployment. This situation proposes a challenge, because we want to integrate an open sensorized platform in a proprietary solution. To address the integration of *SEnviro Things* of the smart campus, GEOEvent extension for ArcGIS Server is used. We will talk about this in the next sections. Also, the next subsection details the way to integrate all *SEnviro Things*. The first subsection presents the architectural overview. The second subsection details how we use GEOEvent. The last subsection provides an in depth explanation of the visualization client; it has been developed to show the data provided by *SEnviro Things* in the *SmartUJI* context.

5.1. Architecture overview

To get a general overview, Figure 8 shows the scheme of the complete infrastructure with all of the system‘s components and describes each of them in different layers. In the first step, *SEnviro Thing* nodes are resided, it is built with an Arduino micro-controller as *SEnviro Core*. Each sensorized platform offers Wi-Fi connection to send the measured observations. We have selected this type of connectivity due to the ubiquitousness of official WI-Fi connection throughout the university campus.

Each *Senviro Thing* is able to adapt to its behaviour, updating the information about which sensor is active and the frequency at which measurements are taken (as detailed in Section 4). At regular intervals, the platform requests a new behaviour to a server using a RestFul service. Then, the *SEnviro Thing* will get a new behaviour following the new configuration received. The sensorized platform sends the observations to a server. To offer this integration a sensor collector service has been created. Different operations are available to collect the observations measured. When one *SEnviro Thing* catches a new observation it invokes these operations to send data to the server using SDM module. When the SDM receives a new observation, it connects with the different adapters to send the data for each service available. To accomplish this objective, we have followed our previous work [53, 12] to connect with different OGC instances, such as SOS and candidate standard SensorThing API. Also, we provide a lighter solution for data sensors access using RestFul interface used in [53]. To offer all these interfaces, we provide different adapters to publish the sensor data in different service instances or databases. To offer a SensorThing interface, we use an external server to test this API, which is offered by the SensorThings API. For SOS, the 52North SOS 4.x implementation (which implements the SOS 2.0 interface standard) is used. In this case, the server is deployed using an Apache Tomcat server and sensor data are stored in a PostgreSQL/PostGIS database[13] following the documented standard installation routines. Finally, for the last

[13]See http://postgis.net/ (last accessed: November, 2015)

interface, a Lightweight RestFul service is implemented using JAX-RS[14] and Jersey[15] and one MongoDB database[16] is used to store the observations. As announced above, to integrate the sensor data from *SEnviro Things* in the *SmartUJI*, we use GEOEvent extension. In order to provide the sensor data from the *SEnviro* platform, GEOEvent connects to the Lightweight RestFul service using polling techniques. GEOEvent has three layers with the sensor data as an output. These layers are offered using publish/subscribe connection and this kind of layer is based on ESRI features in the JSON encoding [17]. In the next section (Section 5.2), we will detail the GEOEvent usage in depth. The last step, is its integration in the smart campus. As commented above, the smart campus is a modular, extensible cartography-based platform to store, access and manage all the data, resources, common services and functionalities required to build a wide range of applications in a smart campus context.

5.2. GEOEvent, as a connector between IoT and smart cities

As we commented, one of the most important challenges of the present work was to integrate an open sensorized platform in smart city development using proprietary software. For this we have chosen to use an extension for ArcGIS server called GeoEvent. GeoEvent is a tool developed by ESRI that enables processing real-time data. This tool can connect to many types of streaming data feeds as inputs and apply different processes over the pro-

[14]See http://jax-rs-spec.java.net/ (last accessed: November, 2015)
[15]See https://jersey.java.net/ (last accessed: November, 2015)
[16]See https://www.mongodb.org/ (last accessed: November, 2015)
[17]See http://www.json.org/ (last accessed: November, 2015)

vided data. These processes can generate alarms or results as output. For instance a GIS layer can be updated, an alarm can be launched when certain events occur or the process results can be served using RestFul or real-time interfaces, like websockets [18]. This ArcGIS extension helps us to integrate data between *SEnviro* and *SmartUJI* platforms. In our work, GEOEvent connects with the lightweight RestFul service (detailed in the Section 4) to adapt the connection between *SEnviro* and *SmartUJI* and retrieves the last observations encoded as JSON. This integration permits using the data provided by the sensorized platform easily. To achieve this, we have configured this connection and a refresh has been applied to obtain the observation in the same frequency that each *SEnviro Thing* produces as a new observation. Whenever GEOEvent receives a new observation, it automatically updates two layers provided as outputs. Figure 9 resumes all the steps followed to connect with GEOEvent.

The GEOEvent process has three different GIS layers as an output. The first layer is a *FeatureLayer* type, we will call this layer as *FeatureLayerSensorData* (FLSD). This layer will be used to offer both historical and current observations. The second layer, called *StreamLayerSensorData* (SLSD), follows the type *StreamLayer* and will be used to publish the latest observations in real-time. These two layers are used to only get observation values. Finally, another *FeatureLayer*, *FeatureLayerSensorLocation* (FLSL), provides the sensor locations. This layer is static and is not updated each time. The *StreamLayer* extends the *FeatureLayer* in order to connect to a stream of data

[18]See https://www.websocket.org/ (last accessed: November, 2015)

using WebSockets. *StreamLayer* connects to a server that emits geographic features continuously. While the *FeatureLayer* is used to map relatively static data, the stream layer is suitable when you would like to map dynamic streams of data that are unbounded and continuous. When a stream layer is added to a map, users are able to see any real-time updates pushed out by the server.

The *StreamLayer* offers some advantages with respect to the *FeatureLayer*. It is more responsive, because features appear on the map right away. Also, the messages are sent without extra headers, so it is a more efficient transfer of data. *StreamLayer* offers better performance than *FeatureLayer*, because websocket connections "can provide a 500:1 or depending on the size of the HTTP headers even a 1000:1 reduction in unnecessary HTTP header traffic and 3:1 reduction in latency" [19]. All the defined layers are stored and backed by an Enterprise Geodatabase (EGDB) offered through one ArcGIS Server instance. These three layers will be used to visualize the sensor data on a web-map or similar.

5.3. SEnviro APP

To complete the *SEnviro Thing* integration, we have proposed a new *SmartUJI* application to access sensor data provided by each *SEnviro Thing*. The Figure 10 represents all applications that make up the current smart campus and *SEnviro APP*. The *SEnviro APP* is a web-application in charge of integrating the sensor data with the existing smart campus infrastructure using the layers provided by the *SmartUJI Core*. The web is the perfect

[19] See http://www.websocket.org/quantum.html (last accessed: November, 2015)

environment for this kind of applications because it is one of the most accessible platforms [54]. In this way, our prototype can be tested on every platform that supports one of the major browsers available (for example, Mozilla Firefox, Google Chrome or derivatives). Our solution for *SmartUJI* is built following the concept of reusable Software [55]. In this way, it is easy for us to create new prototypes for testing new features.

ESRI provides the ArcGIS Javascript API for accessing ESRI services and for creating geo-web applications. Our application is based on this technology. Figure 11 shows a screen-shot about *SEnviro APP*.

The main objective of this web-application is to connect to the different layers provided by the ArcGIS server. The web-application has two main functionalities, one of them is offering the real-time data provided by the *SEnviro Things*. The other is to relate these data the campus buildings and visualizing them.

To achieve this, *SEnviro APP* subscribes to the *StreamLayer* in order to retrieve the last observations measured. To get historical sensor data, *SEnviro APP* it requests all previous observations from the FLSD. Sensor locations are obtained from the FLSD.

The *StreamLayer* with the sensor data carries the sensors values and the identifier of the building where the sensor is located. This information is used to relate sensor data with the buildings *FeatureLayer*. A custom *Renderer* for the buildings *FeatureLayer* is used to show the data over the buildings. In this way, we can implement a *Renderer* depending on the type of sensors we are dealing with. Also, it is possible to adjust the behaviour of the *Renderer* based on the maximum and minimum value of the received data.

The ArcGIS Javascript API has proven to be flexible enough to provide the ability to dynamically change the *Renderer* of a specific Layer. The user can select and filter sensor data using a very simple form (located on the left of the application) and the map loads the sensor's data and sets the correct *Renderer* on the *FeatureLayer*. For this example, each *Renderer* is preloaded into a *RendererFactory* module. In this prototype application, three types of renders are available, to support the sensors: light, temperature and humidity, because these are presented in all *SEnviro Things*. Our system gathers data from many sensors distributed over the campus, but this particular prototype covers only a small subset of them.

The representation is done extending the *ClassBreaksRenderer* provided by the ArcGIS Javascript API. This Render uses some threshold values to define the number of classes the data are divided into. Afterwards, a special symbol is selected for each class. In our prototype, the data are divided into five classes and the values are assigned to: Very Low, Low, Medium, High and Very High (following some defined thresholds). Figure 12 shows the visualization of each sensor that the application currently provides.

SEnviro APP is dynamically updated with the latest observations using SLSD. The FLSD is used when we select a prior date. FLSL is retrieved in order to know the geospatial position of each sensorized platform.

Furthermore, the application is built using the Mobile-first approach. In this way, it can be easily tested on a mobile device. The purpose of this requirement is to show how map-based web application can be implemented to be available in mobile devices. Currently, we are performing usability tests to establish if the application is truly accessible on tablets or smartphones.

Early considerations suggest that a tablet is the preferred mobile device for accessing these kinds of applications. This is mainly because the screen is big enough to show the information on the map at a good scale.

6. Conclusions

We have presented a comprehensive solution to provide IoT devices built with open hardware to the cities of the 21st century. Current cities need these devices in order to know the current state of the city as well as the planning decisions to improve the quality of life of citizens. The proposed solution can measure meteorological and air quality parameters; these phenomena are very useful for decision makers in order to make decisions. Some of these phenomena can be used to determine important indicators. For example, high temperatures can be an indicator of increased electricity consumption. Also, air quality phenomena can be used to analyze traffic conditions or other aspects that have an impact on the atmosphere. To our knowledge only two approaches offer a real-time connection (see Section 2.5), including ours. The current city necessities have to know what is happening at the same moment that it is occurring, for this reason is crucial to offer a real-time communication.Real-time is one of the key features, as there are many situations that require obtaining the data as soon as these have been produced. For example, observations of environmental conditions have to be made as soon as possible to determine any action plan. Other examples could be, the observations of movement of vehicles around a transport network, where data from a network of cameras to be transferred in real time to a central control center to monitor traffic flow and adjust sequences traffic lights and speed limits and automatically administer the penalties for traffic violations.

The third point, faced by the use of open-hardware, and explained in our work can be used to guide developers and policy makers to provide a comprehensive solution for the adaptation of a sensorized platform created with open-hardware within a smart city context. This is taken as a sensorized platform development using hardware and open software, called *SEnviro* [12]. *SEnviro* is integrated within a proposed smart city for a university campus, called *SmartUJI* [13].

The open hardware components usage allows that our approach can be used not only by developers, it can be used by city decision makers and citizens. The citizens can form part of the sensor network following the 'crowdsourcing' or 'participatory sensing' [56]. They can build new *SEnviro* devices and the data provided by these devices can be analyzed in order to make decisions. In this way, citizens can participate as input for governments.

The current cities need has led to the evolution of the traditional WSNs to "smart networks", permanently connected to the Internet (IoT), offering the possibility to know what is happening at every moment. This term has become very popular in the media and business realms. Therefore, it has triggered many business models, with closed solutions (proprietary implementation) for monitoring and controlling different city environments. These solutions have the disadvantage of not being able to be used openly and not having total control over them [57]. As such, another movement has emerged to solve this problem; it is the open-hardware movement. Some studies have demonstrated that open-hardware projects are equally as valid as proprietary solutions [21], in addition open-hardware solutions are much cheaper than the proprietary ones [58]. If we see the features and greater advantages that open hardware solutions offer over proprietary ones, then we can observe how open source devices could be a possible solution for the implementations of the IoT.

Finally, the last goal of our work is the integration between the sensorized platform within the platform of a smart campus following the paradigms of CC and M2M. In this integration, GIScience takes importance since it offers different standards to interconnect the "smart things" in an interoperable and scalable way. For this reason, GIScience becomes a good candidate to use in the future development of smart cities. Also, the methodology defined a possible way to connect an open platform within a platform of smart city using proprietary source software. It has been reached successfully, so it guarantees the scalability and interoperability of the *SmartUJI*.

As future work, increased deployment of sensors is proposed to improve the granularity of the web-maps obtained and increase their quality. The generalization of the adaptation of *SEnviro* platform that can be deployed in other developments of different smart cities is also contemplated. Another open issue is to analyze the quality of the data provided by the sensors and make different comparative between *SEnviro* sensors and professional sensor stations. Finally, another future line that we pursue is the real-time analysis, in order to detect anomalies over the data obtained from sensors for decision-making [59].

[1] G. K. Heilig, World urbanization prospects: the 2011 revision, United Nations, Department of Economic and Social Affairs (DESA), Popula-

tion Division, Population Estimates and Projections Section, New York (2012).

[2] D. Satterthwaite, Cities' contribution to global warming: notes on the allocation of greenhouse gas emissions, Environment and urbanization 20 (2008) 539–549.

[3] R. Moss Kanter, S. S. Litow, Informed and interconnected: A manifesto for smarter cities, Harvard Business School General Management Unit Working Paper (2009).

[4] D. INFSO, Networked enterprise & rfid infso g. 2 micro & nanosystems, in co-operation with the working group rfid of the etp eposs, internet of things in 2020, roadmap for the future [r], Information Society and Media, Tech. Rep (2008).

[5] A. Kramers, M. Höjer, N. Lövehagen, J. Wangel, Smart sustainable cities–exploring ict solutions for reduced energy use in cities, Environmental Modelling & Software 56 (2014) 52–62.

[6] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Pichler-Milanovic, E. Meijers, Smart cities-Ranking of European medium-sized cities, Technical Report, Vienna University of Technology, 2007.

[7] D. Uckelmann, M. Harrison, F. Michahelles, An architectural approach towards the future internet of things, in: Architecting the internet of things, Springer, 2011, pp. 1–24.

[8] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things

(iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (2013) 1645–1660.

[9] S. Dirks, M. Keeling, J. Dencik, Ibm how smart is your city? helping cities measure progress, Global Business Services Government Executive Report IBM Institute for Business Value Copyright IBM Corporation (2009).

[10] K. Crowston, K. Wei, J. Howison, A. Wiggins, Free/libre open-source software development: What we know and what we do not know, ACM Computing Surveys (CSUR) 44 (2012) 7.

[11] Official arduino website, http://www.arduino.cc, ???? Accessed: 2014-11-3.

[12] S. Trilles, A. Luján, Ó. Belmonte, R. Montoliu, J. Torres-Sospedra, J. Huerta, Senviro: A sensorized platform proposal using open hardware and open standards, Sensors 15 (2015) 5555–5582.

[13] J. Huerta, Smartuji: Integrating information, services, and participation, Technical Report, Specialist Meeting Advancing the Spatially Enabled Smart Campus, 2013.

[14] P. Bellavista, G. Cardone, A. Corradi, L. Foschini, Convergence of manet and wsn in iot urban scenarios, IEEE Sensors Journal 13 (2013) 3558–3567.

[15] N. Walravens, P. Ballon, Platform business models for smart cities: from control and value to governance and public value, IEEE Communications Magazine 51 (2013) 72–79.

[16] C. Manville, G. Cochrane, J. Cave, J. Millard, J. K. Pederson, R. K. Thaarup, A. Liebe, M. Wissner, R. Massink, B. Kotterink, Mapping smart cities in the eu (2014).

[17] P. Rashidi, D. J. Cook, L. B. Holder, M. Schmitter-Edgecombe, Discovering activities to recognize and track in a smart environment, IEEE transactions on knowledge and data engineering 23 (2011) 527–539.

[18] D. J. Cook, M. Youngblood, S. K. Das, A multi-agent approach to controlling a smart environment, in: Designing smart homes, Springer, 2006, pp. 165–182.

[19] M. Zhang, A. Ishii, N. Nishiyama, S. Matsumoto, T. Ishii, Y. Yamasaki, K. Kataoka, Pegylated calcium phosphate nanocomposites as smart environment-sensitive carriers for sirna delivery, Advanced Materials 21 (2009) 3520–3525.

[20] V. R. Jakkula, D. J. Cook, Using temporal relations in smart environment data for activity prediction, in: Proceedings of the 24th International conference on machine learning, pp. 20–24.

[21] R. Fisher, L. Ledwaba, G. Hancke, C. Kruger, Open hardware: a role to play in wireless sensor networks?, Sensors 15 (2015) 6818–6844.

[22] V. M. Rohokale, N. R. Prasad, R. Prasad, A cooperative internet of things (iot) for rural healthcare monitoring and control, in: Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, IEEE, pp. 1–6.

[23] C. Doukas, I. Maglogiannis, Bringing iot and cloud computing towards pervasive healthcare, in: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, IEEE, pp. 922–926.

[24] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, G. Marrocco, Rfid technology for iot-based personal healthcare in smart spaces, Internet of Things Journal, IEEE 1 (2014) 144–152.

[25] M. Schneps-Schneppe, D. Namiot, A. Maximenko, D. Malov, Wired smart home: energy metering, security, and emergency issues, in: Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on, IEEE, pp. 405–410.

[26] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, X. Lin, Smart community: an internet of things application, Communications Magazine, IEEE 49 (2011) 68–75.

[27] N. Kumar, J.-H. Lee, J. J. Rodrigues, Intelligent mobile video surveillance system as a bayesian coalition game in vehicular sensor networks: learning automata approach (2014).

[28] P. Hank, S. Müller, O. Vermesan, J. Van Den Keybus, Automotive ethernet: in-vehicle networking and smart mobility, in: Proceedings of the Conference on Design, Automation and Test in Europe, EDA Consortium, pp. 1735–1739.

[29] A. Somov, C. Dupont, R. Giaffreda, Supporting smart-city mobility with

cognitive internet of things, in: Future Network and Mobile Summit (FutureNetworkSummit), 2013, IEEE, pp. 1–10.

[30] A. Botta, W. de Donato, V. Persico, A. Pescapé, On the integration of cloud computing and internet of things, in: Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, IEEE, pp. 23–30.

[31] A. Weiss, Computing in the clouds, Computing 16 (2007).

[32] X. Fang, S. Misra, G. Xue, D. Yang, Managing smart grid information in the cloud: opportunities, model, and applications, IEEE network 26 (2012) 32–38.

[33] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, R. Morris, Smarter cities and their innovation challenges, Computer 44 (2011) 32–39.

[34] S. K. Dash, S. Mohapatra, P. K. Pattnaik, A survey on applications of wireless sensor network using cloud computing, International Journal of Computer science & Engineering Technologies (E-ISSN: 2044-6004) 1 (2010) 50–55.

[35] G. C. Fox, S. Kamburugamuve, R. D. Hartman, Architecture and measured characteristics of a cloud based internet of things, in: Collaboration Technologies and Systems (CTS), 2012 International Conference on, IEEE, pp. 6–12.

[36] C. Potter, G. P. Hancke, B. Silva, Machine-to-machine: Possible applications in industrial networks, in: Industrial Technology (ICIT), 2013 IEEE International Conference on, IEEE, pp. 1321–1326.

[37] K. C. Clarke, Getting started with geographic information systems, volume 3, Prentice Hall Upper Saddle River, NJ, 1997.

[38] J. J. Rodrigues, P. A. Neves, A survey on ip-based wireless sensor network solutions, International Journal of Communication Systems 23 (2010) 963–981.

[39] J. Galache, J. R. Santana, V. Gutiérrez, L. Sánchez, P. Sotres, L. Muñoz, et al., Towards experimentation-service duality within a smart city scenario, in: Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on, IEEE, pp. 175–181.

[40] G. Merlino, D. Bruneo, S. Distefano, F. Longo, A. Puliafito, Stack4things: integrating iot with openstack in a smart city context, in: Smart Computing Workshops (SMARTCOMP Workshops), 2014 International Conference on, IEEE, pp. 21–28.

[41] M. Faisal, S. Bakar, P. Rudati, The development of a data acqusition system based on internet of things framework, in: ICT For Smart Society (ICISS), 2014 International Conference on, IEEE, pp. 211–216.

[42] Á. Asensio, T. Blanco, R. Blasco, Á. Marco, R. Casas, Managing emergency situations in the smart city: The smart signal, Sensors 15 (2015) 14370–14396.

[43] R. Jalali, K. El-khatib, C. McGregor, Smart city architecture for community level services through the internet of things, in: Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on, IEEE, pp. 108–113.

[44] A. Zanella, M. Zorzi, Applying internet of things paradigm to smart city: communication model and experimentation, E-LETTER (2014).

[45] B. Pokric, S. Krco, D. Drajic, M. Pokric, I. Jokic, M. J. Stojanovic, ekonet-environmental monitoring using low-cost sensors for detecting gases, particulate matter, and meteorological parameters, in: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on, IEEE, pp. 421–426.

[46] J. Torres-Sospedra, J. Avariento, D. Rambla, R. Montoliu, S. Casteleyn, M. Benedito-Bordonau, M. Gould, J. Huerta, Enhancing integrated indoor/outdoor mobility in a smart campus, International Journal of Geographical Information Science (2015) 1–14.

[47] Y.-S. Chang, H.-D. Park, Xml web service-based development model for internet gis applications, International Journal of Geographical Information Science 20 (2006) 371–399.

[48] I. Masser, GIS worlds: creating spatial data infrastructures, volume 338, ESRI press Redlands, CA, 2005.

[49] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, J. Huerta, Comprehensive analysis of distance and similarity measures for wi-fi

fingerprinting indoor positioning systems, Expert Systems with Applications 42 (2015) 9263–9278.

[50] D. Guinard, V. Trifa, T. Pham, O. Liechti, Towards physical mashups in the web of things, in: Networked Sensing Systems (INSS), 2009 Sixth International Conference on, IEEE, pp. 1–4.

[51] A. Tamayo, P. Viciano, C. Granell, J. Huerta, Empirical study of sensor observation services server instances, in: Advancing geoinformation science for a changing world, Springer, 2011, pp. 185–209.

[52] M. Botts, G. Percivall, C. Reed, J. Davidson, Ogc® sensor web enablement: Overview and high level architecture, in: GeoSensor networks, Springer, 2008, pp. 175–190.

[53] S. Trilles, O. Belmonte, L. Diaz, J. Huerta, Mobile access to sensor networks by using gis standards and restful services, Sensors Journal, IEEE 14 (2014) 4143–4153.

[54] T. Berners-Lee, Www: past, present, and future, Computer 29 (1996) 69–77.

[55] A. Jatain, T. Chawla, Architecture for developing reusable components for gis systems, in: Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on, pp. 482–485.

[56] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava, Participatory sensing, Center for Embedded Network Sensing (2006).

[57] S. Shrestha, J. Moore, J. A. Nocera, Open-source platform: exploring the opportunities for offline mobile learning, in: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, ACM, pp. 653–658.

[58] A. Powell, Democratizing production through open source knowledge: from open software to open hardware, Media, Culture & Society 34 (2012) 691–708.

[59] S. Trilles, S. Schade, Ó. Belmonte, J. Huerta, Real-time anomaly detection from environmental data streams, in: AGILE 2015, Springer International Publishing, 2015, pp. 125–144.

875 ## List of figures

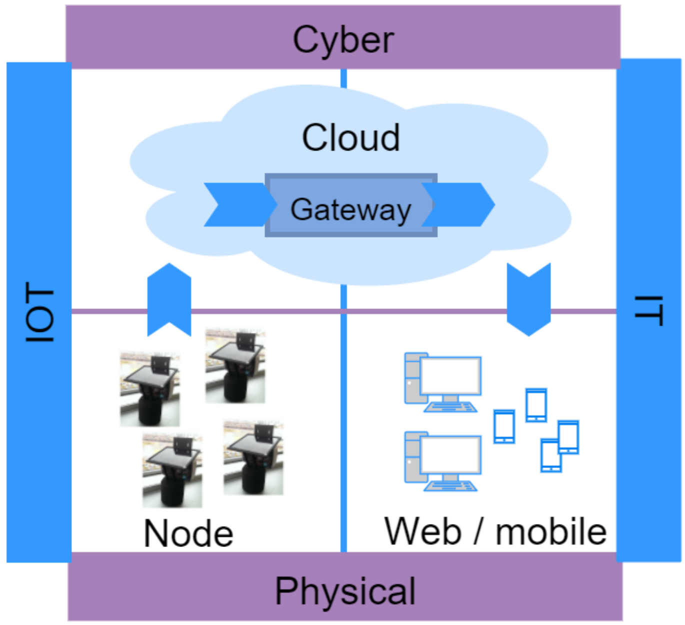


Figure 1: Representation of the cycle provided by IoT and IT, that it is divided between the physical and cyber world.

Figure 2: Screen-shot about the *SmartUJI* web.

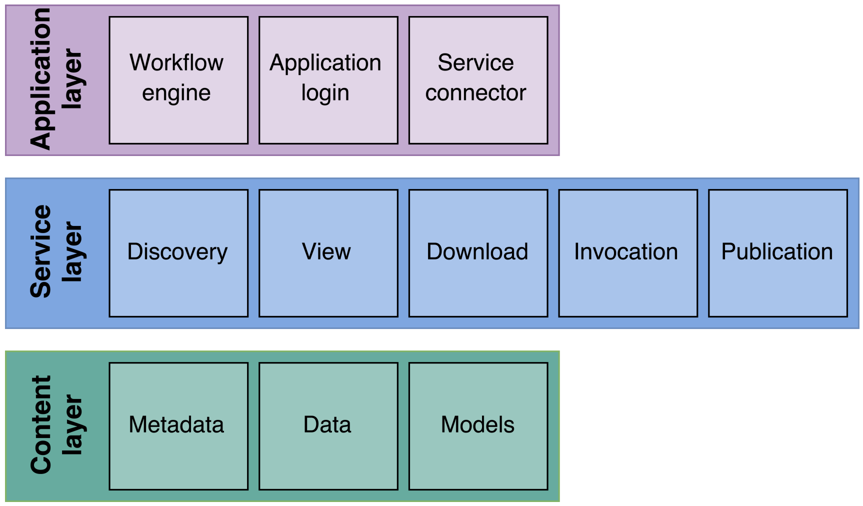
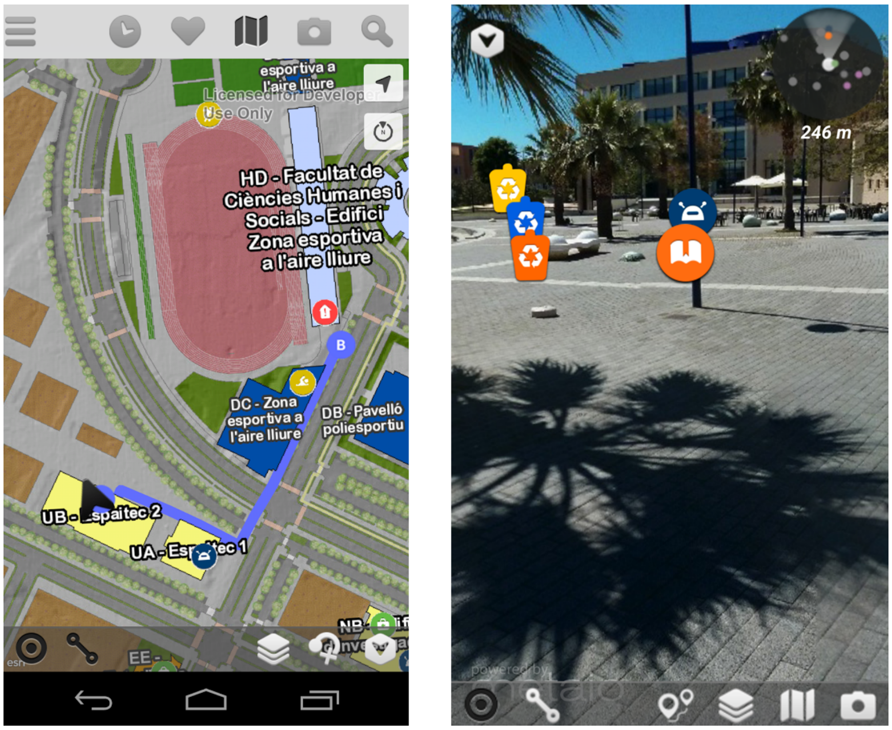


Figure 3: Typical SDI three layers architecture applied in *SmartUJI*.



a) b)

Figure 4: Two screen-shots about the SmartUJI APP (a) and SmartUJI AR (b).

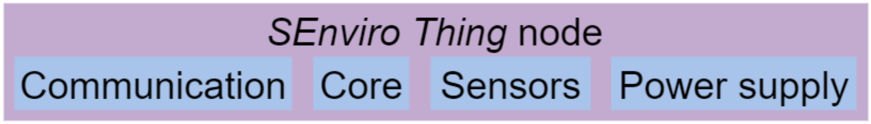


Figure 5: *SEnviro Thing* parts.

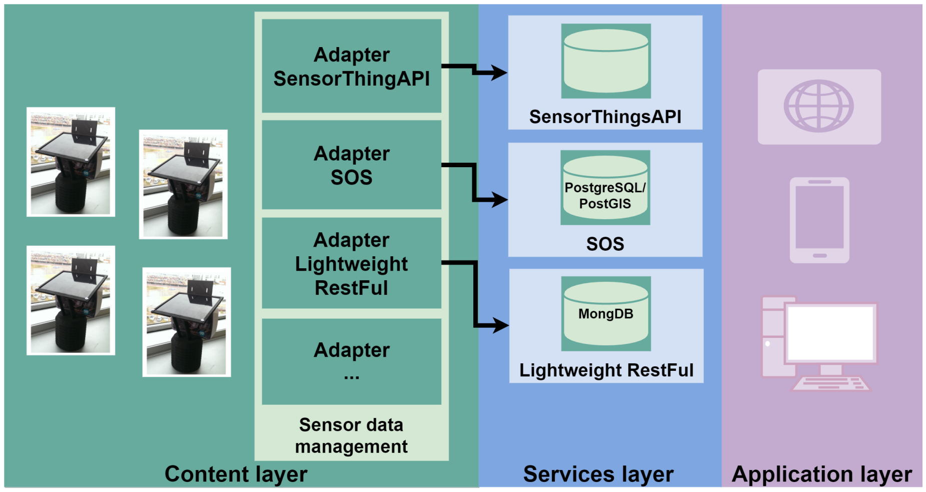


Figure 6: General overview about the *SEnviro* work-flow following three layers architecture.

Figure 7: Example of *SEnviro Thing* installed in Espaitec 2 building.

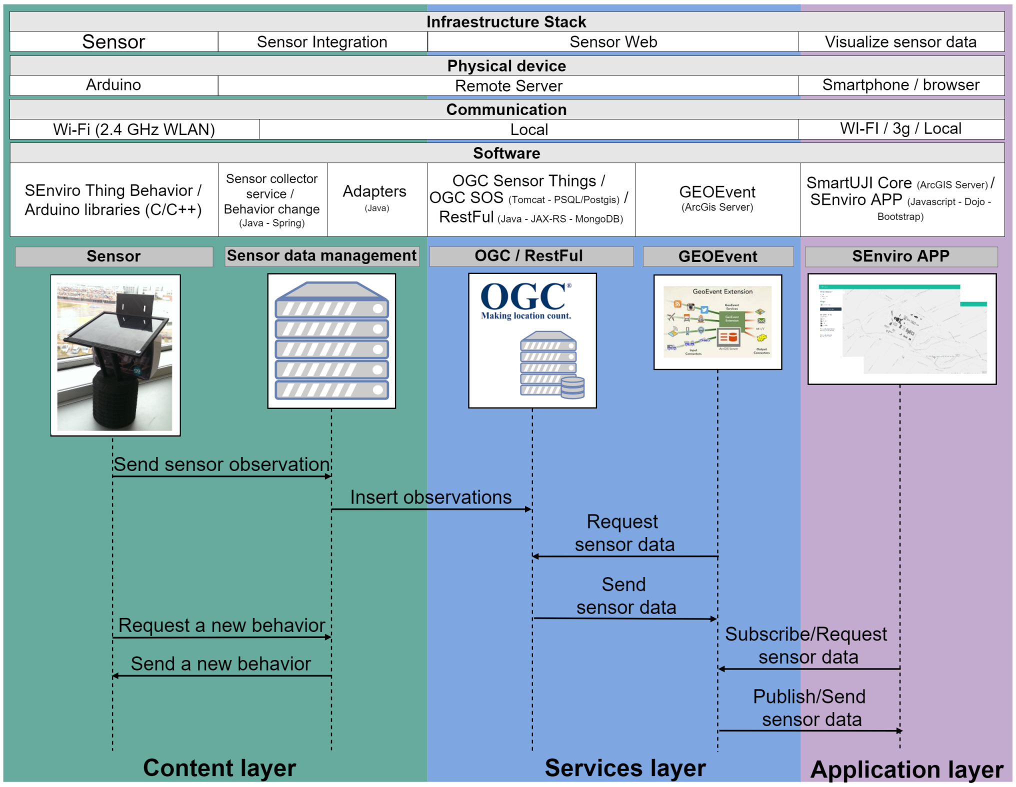


Figure 8: Sequence diagram of the processing of an exemplary usage of the system (lower half). The upper half gives information about the realization of the different components of the infrastructure.

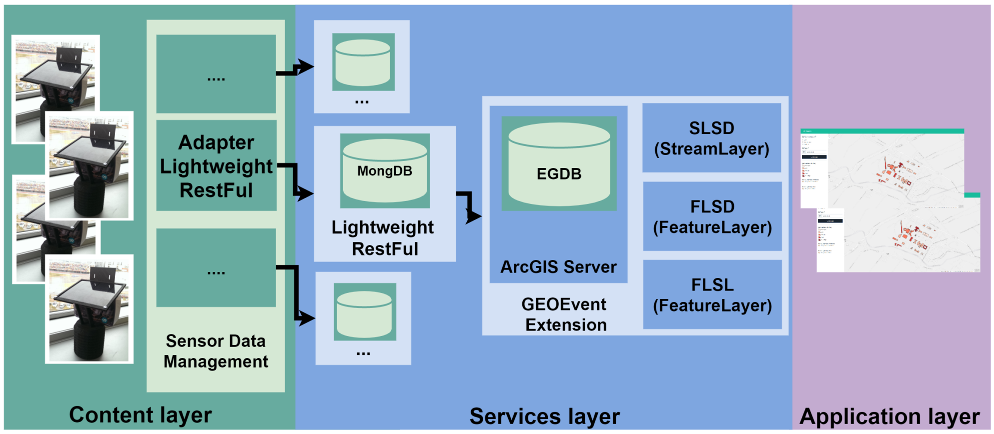


Figure 9: General overview about the integration between *SEnviro* and *SmartUJI* using GEOEvent following three layers architecture.

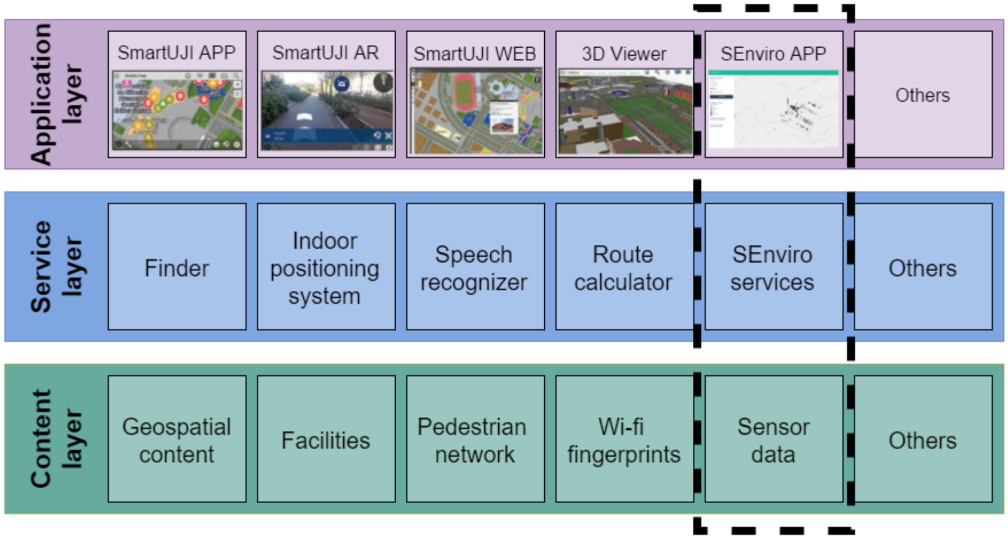


Figure 10: Proposed extended *SmartUJI* architecture to add *SEnviro* platform.

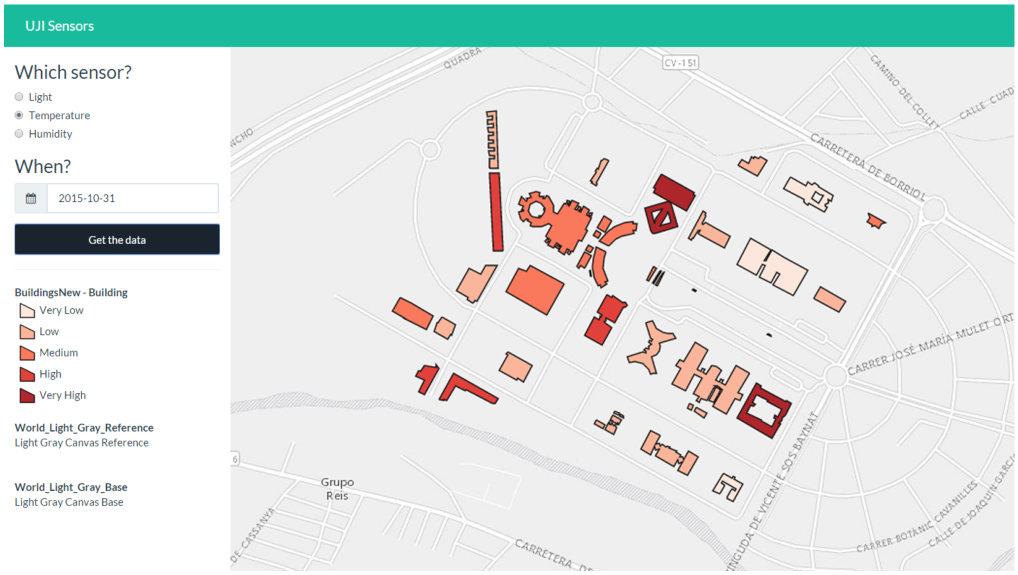


Figure 11: Screen-shot of the application developed that use *SmartUJI* services to visualize the data provide by the *SEnviro Thing*.

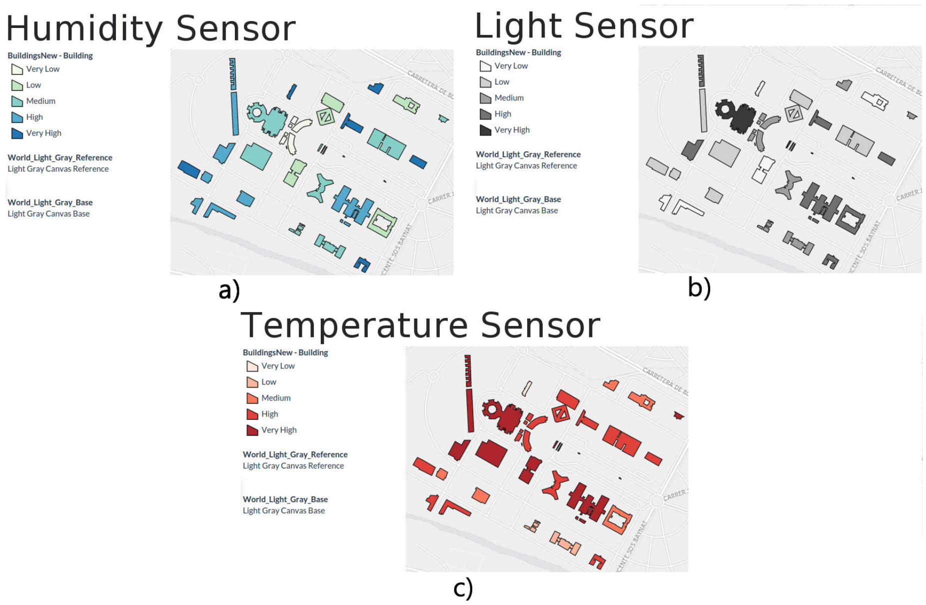


Figure 12: SEnviro Application Sensors visualization. It shows some of the sensors the application currently supports a) Humidity b) Light and c) Temperature.