Table of Contents

International Journal on Semantic Web and Information Systems

Volume 12 • Issue 4 • October-December-2016 • ISSN: 1552-6283 • eISSN: 1552-6291
An official publication of the Information Resources Management Association

Special Issue on Semantic Challenges for the Variety and Velocity Dimensions of Big Data

Research Articles

COPYRIGHT

The *International Journal on Semantic Web and Information Systems* is indexed or listed in the following: ACM Digital Library; Bacon's Media Directory; Burrelle's Media Directory; Cabell's Directories; Compendex (Elsevier Engineering Index); CSA Illumina; Current Contents®/Engineering, Computing, & Technology; DBLP; DEST Register of Refereed Journals; Gale Directory of Publications & Broadcast Media; GetCited; Google Scholar; INSPEC; Journal Citation Reports/Science Edition; JournalTOCs; Library & Information Science Abstracts (LISA); MediaFinder; Norwegian Social Science Data Services (NSD); Science Citation Index Expanded (SciSearch®); SCOPUS; The Index of Information Systems Journals; The Standard Periodical Directory; Thomson Reuters; Ulrich's Periodicals Directory; Web of Science

International Journal on Semantic Web and Information Systems
Volume 12 • Issue 4 • October-December 2016

Managing Large Amounts of Data Generated by a Smart City Internet of Things Deployment

Jorge Lanza, Universidad de Cantabria, Santander, Spain
Pablo Sotres, Universidad de Cantabria, Santander, Spain
Luis Sánchez, Universidad de Cantabria, Santander, Spain
Jose Antonio Galache, Universidad de Cantabria, Santander, Spain
Juan Ramón Santana, Universidad de Cantabria, Santander, Spain
Verónica Gutiérrez, Universidad de Cantabria, Santander, Spain
Luis Muñoz, Universidad de Cantabria, Santander, Spain

ABSTRACT

The Smart City concept is being developed from a lot of different axes encompassing multiple areas of social and technical sciences. However, something that is common to all these approaches is the central role that the capacity of sharing information has. Hence, Information and Communication Technologies (ICT) are seen as key enablers for the transformation of urban regions into Smart Cities. Two of these technologies, namely Internet of Things and Big Data, have a predominant position among them. The capacity to "sense the city" and access all this information and provide added-value services based on knowledge derived from it are critical to achieving the Smart City vision. This paper reports on the specification and implementation of a software platform enabling the management and exposure of the large amount of information that is continuously generated by the IoT deployment in the city of Santander.

KEYWORDS

Big Data, Internet of Things, Platform, Smart City

1. INTRODUCTION

The smart city concept has undergone a rapid growth in popularity and interest recently. The fact that in the future the majority of the world's population will live in cities (the World Health Organization (2010) has predicted that by 2050 seventy percent of the world's population will live in urban areas) and that more than half of the world's population already lives in cities encourages researchers and managers to look for new solutions guaranteeing the sustainability and efficiency of such complex ecosystems.

Different city-domain stakeholders (technicians, city planners, politicians, researchers, etc.) will need to take measures aimed at assuring that some key quality criteria related to the sustainability and efficiency in the city domain are fulfilled. Thus, many different approaches are being followed to achieve the smart city vision (Belissent, J., 2010), (Zygiaris, S., 2013), (Nam, T., & Pardo, T. A., 2011). Something that is common to all these approaches is the central role that the capacity of

DOI: 10.4018/IJSWIS.2016100102

sharing information has. Hence, Information and Communication Technologies (ICT) are seen as key enablers for the transformation of urban regions into smart cities. Two of these technologies, namely Internet of Things and Big Data, have a predominant position among all of them (Vilajosana et al., 2013). The capacity to "sense the city" and access all this information to provide added-value services based on the knowledge derived from it are critical to achieving the smart city vision. This revolution is still only in its infancy as suitable infrastructures are being deployed and significant investments are being made in city infrastructures.

Among the different challenges that must be considered when dealing with a real IoT deployment summarized by Gluhak et al. (2011) and Lanza et al. (2015), the scale, heterogeneity and data-centricity aspects are the focus of the work presented in this paper. This paper's main contribution is the specification and implementation of the software platform enabling the management and exposure of the large amount of information that is continuously generated by a real large-scale IoT deployment.

The remainder of the paper is structured as follows. In order to fully understand the scale and heterogeneity aspects addressed by the platform presented in the paper, Section 2 will summarize the main details of the infrastructure deployed. This description will present the general scenario that sets the framework for the work presented in this paper. The big data facet of the deployment is highlighted. Better knowledge of the infrastructure will help us to understand the key design considerations, which will also be presented in this section. Section 3 will present a review of related work. This review will centre on analyzing analogous deployments of IoT infrastructures focusing on the smart city domain and placing emphasis on the software platforms handling the data they generate. The detailed description of the platform implemented (main contribution of the paper) will be introduced in Section 4. Last but not least, Section 5 will conclude the paper, highlighting the main contributions and discussing the key issues raised throughout the paper.

2. LARGE SCALE SMART-CITY IOT DEPLOYMENT

As has already been mentioned, the basis for the main contribution of this paper, namely the description of the specification and implementation of the software platform enabling the management and exposure of a large amount of IoT information, is a real city-scale IoT deployment. This section will present an overview of this IoT infrastructure which is deployed in the city of Santander.

The insights into this deployment and details of the installed devices have already been described by Sanchez et al., 2014; and Lanza et al., 2015. However, this paper will extend these by introducing detailed information on the data generation aspect. Both the description of the infrastructure and the analysis of the data generation (bulk amount and patterns) are important for helping in the understanding of the data management requirements. These requirements are also mentioned in this section. Finally, this description is also relevant for gauging the challenges faced by the implemented software platform, described in Section 4.

2.1. SmartSantander General Framework

The SmartSantander project (SmartSantander, 2010) targeted the creation of a European experimental test facility for research and experimentation on architectures, key enabling technologies, services and applications for the Internet of Things (IoT) in the context of a smart city. The SmartSantander platform includes a continuously growing Internet of Things (IoT) infrastructure spread throughout the city that currently encompasses more than 12,000 diverse IoT devices (fixed and mobile sensor nodes, NFC tags, gateway devices, citizens' smartphones, etc.). This facility aims to leverage key

IoT-enabling technologies and to provide the research community with a unique-in-the-world platform for large-scale IoT experimentation and evaluation under real-world operational conditions.

However, this testbed goes beyond the experimental validation of novel IoT technologies. It also aims at supporting the assessment of the socio-economical acceptance of new IoT solutions and the quantification of service usability and performance with end users in the loop.

2.2. Deployed IoT Infrastructure Overview

The IoT experimentation facility deployed in Santander was selected using a cyclic approach. The deployment, influenced by Santander Municipality's strategic smart-city service requirements, intentionally provided a concentration of IoT devices in the city centre (a 1 Km^2 area) in order to achieve the maximum possible impact on the citizens. Nonetheless, other city areas are also covered.

Figure 1 shows an excerpt view of the deployment. The different markers represent the deployed nodes (e.g. illuminance, sound pressure level, ambient temperature, mobile nodes or car presence detection sensors).

One of the main aims of the deployed infrastructure is supporting the analysis and usability and performance quantification of new IoT-based added-value services with end users in the loop. To attract the widest interest and demonstrate its usefulness, the deployment of the IoT experimentation infrastructure has been undertaken to create interesting use-cases that will generate an impact. In this respect, application areas have been selected based on their high potential impact on the citizens. Diversity, dynamics and scale of the IoT environment are also taken into consideration in the selection of application use cases.

The SmartSantander testbed is composed of around 3000 IEEE 802.15.4 devices, 200 GPRS modules and 2000 joint NFC tag/QR code labels deployed both at static locations (streetlights, facades, bus stops) as well as on-board mobile vehicles (buses, taxis). Moreover, smartphones belonging to citizens who have downloaded the *Pulso de la Ciudad App* (Pulso de la Ciudad, 2013) are also part of the testbed infrastructure.

Over the deployed testbed, several use cases have been implemented:

- **Environmental Monitoring:** Around 2000 IoT devices installed (mainly in the city centre), on lampposts and facades provide measurements of different environmental parameters, such as temperature, CO, noise, light and car presence);

Figure 1. Santander IoT infrastructure deployment excerpt view

- **Mobile Environmental Monitoring:** In order to extend the aforementioned environmental monitoring use case, apart from measuring at static points, devices located in vehicles retrieve environmental parameters (related to air pollution). Sensors are installed in 150 public vehicles, including buses, taxis and police cars;
- **Outdoor Parking:** Almost 400 parking sensors (based on ferromagnetic technology), buried under the asphalt, have been installed in the main parking areas of the city centre in order to detect parking site availability in these zones;
- **Guidance to free parking lots:** Taking information retrieved by the deployed parking sensors, 10 panels have been installed at the main streets' intersections in order to guide drivers towards the available free parking lots;
- **Traffic Intensity Monitoring:** Around 60 devices have been deployed at the main entrances to the city of Santander to measure the main traffic condition parameters, such as traffic volumes, road occupancy, and vehicle speed or queue length;
- **Parks and gardens irrigation:** In order to make irrigation as efficient as possible, around 50 devices have been deployed in two green zones of the city to monitor irrigation-related parameters such as moisture temperature and humidity, rain precipitation or wind conditions;
- **NFC/QR tags:** More than 2000 tags have been deployed and distributed throughout different strategic locations. These tags are mainly at transportation points (bus stops, taxi ranks, etc.), points of interest (monuments, etc.) and shops. All the information provided is online and can be updated at any time. Every time one of these tags is read, an observation is generated including information relating to the reader;
- **Participatory sensing:** In this scenario, mobile phones are used as sensing devices automatically feeding information from the device's built-in sensors such as GPS, compass, noise or temperature into the SmartSantander platform. Users can also participate by manually reporting events or incidences occurring in the city, which will subsequently be propagated to other users who are subscribed to the respective type of events.

2.3. Smart City Deployment Data Generation Analysis

It goes without saying that such a deployment generates an enormous amount of data that must be handled, stored and made available. This section will summarize the way in which this data is generated and some rough figures on the overall dimension of the data to be handled.

2.3.1. Data Generation Patterns

First of all, it is important to distinguish between the data generation patterns used by the sensing devices deployed. The data generation pattern for each device is mainly defined by the service it is meant for. Basically, it is possible to identify two patterns: 1) periodic observation generation; and 2) event-based observation generation.

IoT devices programmed with the periodic observation generation pattern will report an observation containing the sensed information on a configurable frequency basis. For fixed devices this frequency is in the time domain, but for mobile devices, it is possible to configure them to work on a specific time or distance period (or a combination of these). Devices using this pattern are employed for the Environmental Monitoring, Traffic Intensity Monitoring and Parks and gardens irrigation services. The observations from the smartphone built-in sensors used in the Participatory Sensing service are also used in this approach.

IoT devices engaged in Outdoor Parking, however, work in an event-based manner. Their observations are only reported upon the detection of a change in the parameter they are monitoring. The part of the Participatory Sensing service where citizens are allowed to report incidences as they encounter them also follows this event-based observation generation pattern. Moreover, observations generated upon the citizens' interaction with any of the QR/NFC tags deployed all over the city are also mapped in this pattern.

2.3.2. Amount of Data Generated

One of the objectives of the SmartSantander platform was to support advanced IoT experimentation. In this sense, since the reporting period for those devices implementing the periodic observation generation pattern is configurable, the decision made in order to address the above-mentioned objective was to establish a high frequency. Taking into consideration just the service needs, the selected frequency leads to an oversampling situation. However, this enabled wider experimentation, and challenged the implemented IoT platform as an even larger infrastructure would have done. For the majority of devices, the time period used for reporting new observations was set to five minutes. For those devices using the event-based observation generation pattern, the number of observations reported depends only on the actual usage of the service. Table 1 summarizes the number of observations generated on a daily basis within the SmartSantander testbed.

It is important to mention that these figures are average values calculated from the total number of observations gathered during March 2014.

2.4. Large Scale IoT Infrastructure Data Management Requirements

Taking into account the deployed infrastructure's characteristics and before describing the system adopted, it is important to present the key aspects that have been addressed during the definition of the IoT data management platform described in this paper. It is also important to highlight that all of the following features have been derived as a result of insightful rounds of external users consuming information generated in the smart city deployment and challenging extensions with additional sources of information:

- **Heterogeneity:** Supporting a wide range of information implies a high level of heterogeneity both in terms of the data managed and in terms of the usage of this data. For data to be useful in a shared utility model it is necessary for them to be well and consistently described. A lot of time and effort is frequently expended in analytical systems in 'cleansing and aligning' data to be able to make data from different sources integrateable and useful. Thus, any IoT data management platform should homogenize this information as it arrives at the system in order to serve it already aligned to a consistent data model. In order to allow a wide range of scenarios and requirements to be addressed, this model should be extendable. Moreover, this model should be conceived in such a way that complex filtering during data access does not have a deep impact on the system performance;
- **Experimentation realism:** Live testbeds intrinsically provide a degree of experimentation realism that even the most detailed simulation cannot achieve (Nordström et al. 2007) but it is also necessary to leverage current software development practices for the accessibility of the testbed functionalities so that the platform is useful for the faster maturation of IoT solutions for

Table 1. Number of observations generated daily in the SmartSantander testbed

Service	Daily Observations
Environmental Monitoring	139,370
Parks and Gardens Irrigation	8,365
Mobile Environmental Monitoring	82,726
Parking Occupancy	13,489
Traffic Management	54,720
Participatory Sensing	6,352
Augmented Reality	1,489

the mass market. In this sense, creating a construct to manage massive volumes of aligned and managed data loses its sense if it is not clear how an application – or a user – gains access to it;

- **Scalability:** Real-world experimentation in a target deployment environment also requires experimentation on an adequate scale. While smaller-scale testbeds with populations of tens up to hundreds of nodes were sufficient for most WSN experiments, IoT experimentation demands a scale of an order of magnitude larger. In order to facilitate access to the information generated by thousands of IoT experimentation nodes, it is necessary to deploy adequate mechanisms which can scale up and allocate access to progressively growing infrastructures;
- **Interoperability and Federation:** Modelling of information is not only required to efficiently handle heterogeneous sources of information and information queries but also to guarantee extendibility of the platform and support for plug and play incorporation of new IoT devices or even legacy datasets and data-streams. In this sense, it is necessary to establish the means by which other infrastructure providers (anyone with a deployment of physical or virtual sensors) federate with the system and enlarge the catalogue of information;
- **Metadata:** Linking the information with the information provider and supporting the metadata concept is of utmost importance when dealing with heterogeneous data consumption needs. For the IoT data management platform to serve the widest variety of users possible it must be possible to apply semantics to the information starting from location and timestamp while also enabling connection to other information attributes such as precision, range, etc.;
- **Security:** In successful IoT technology deployments, the criticality and value of data collected and the importance of control functions mean that it is essential to secure these assets in order to protect from anything from data theft to invasion of privacy. A further challenge is that the security which should be applied is usually highly contextual, depending on identity, application, device, location, time, and potentially other factors. More importantly, when retrieving data from the IoT platform the data set returned is filtered according to the corresponding security policy. If someone is allowed to see historic temperature sensor readings for common areas in a building, but not for other tenants' apartments, then a query to show historic values for the entire building, will be returned only with the relevant data;
- **Ease of Application Development:** Open APIs, conceptually simple and consistent REST interfaces, aligned data are enablers that provide an IoT data management platform with a quick and effective tool for developers to integrate with or develop against;
- **Near Real-Time:** A very common requirement in IoT and smart city applications is to 'get the right data to the right people and place at the right time'. Publish-subscribe is a well-understood paradigm, but it is of utmost importance to support server-level filtering of results, leveraging both the location hierarchy and information type demands. Moreover, this should be done providing a high degree of symmetry between data access and subscription filtering which is highly useful for certain application patterns. More importantly, this means that applications do not have to do the work of filtering and discarding unnecessary data themselves, and where filtering requirements for multiple subscribers are common, economies of scale can be exploited;
- **Historical information:** The scope of 'big data' analytics is a complex and expanding field, with general purpose techniques competing with more focused and specialized solutions addressing the requirements of vertical domains. However, to identify data patterns of significance to operators and providers and to enable optimization of algorithms and services delivery, it is first necessary to provide the means to capture, store and efficiently make available significant amounts of high-quality history data.

3. RELATED WORK

Kitchin (2014) broadly divided the term Smart City into two distinct categories. Whether it is understood as a kind of 'everyware', that is, pervasive and ubiquitous computing and digitally instrumented devices built into the very fabric of urban environments as stated by Greenfield (2006), Hanke et al. (2013) or Schaffers et al. (2011) or as an ecosystem for the development of a new knowledge-based economy (Kourtit et al., 2012), they both share a common prioritisation of data capture and analysis as a means for realising their overall smart city vision.

However, very few real smart city deployments are nowadays available to actually assess data management challenges that arise from such a vision. Bakici et al. (2013) presented the case of Barcelona where a plethora of different smart city initiatives has been produced. However, while in his work a unified strategy for the city is presented, the different initiatives are mostly isolated from each other, thus not achieving the necessary critical mass in terms of generated data for mimicking a holistic smart city scenario, although an Open Data approach is described. Oulu is another well-known example of the deployment of ICT technologies (Ojala et al. 2010) to improve efficiency and well-being in the city. Gil-Castineira et al. (2011) summarized some of the experiments carried out. In all the cases, infrastructure is used to support service provision for the citizens but there is no scheme adopted for managing the information generated. There are plenty of planned pilot projects of smart cities under development that are worth mentioning. The development by Songdo (Strickland 2011) of a fully ubiquitous 1,500 acre city in South Korea started in 2001 and is expected to be complete by 2015, the PlanIT Valley (Carvalho and Campos, 2013) in Northern Portugal, another pilot project for a fully ubiquitous city built from the ground up, scheduled for 2015, and the Fujisawa Sustainable Smart Town (Panasonic Fujisawa SST, Online) have not yet actually gone into a tangible large-scale deployment. In Malaga, another smart city initiative in Spain, the focus has been put on the smart grid aspects (Carillo-Aparicio et al., 2013) with the aim of achieving a reduction in the carbon dioxide emissions. While the current deployment already includes more than 17,000 deployed smart meters, data flows are tailored for each of the implemented use cases thus reducing the challenges regarding the management of data mainly from the variety and heterogeneity viewpoints.

Focusing on the software platforms which may be horizontal to the potential application domains of IoT, interesting related approaches can be found in the literature. Mitton et al. (2012) bring together the cloud and IoT concepts in order to move towards a Cloud of Things (CoT) concept that can abstract sensing data and add new sensing capabilities with no configuration. Whereas the concept is certainly aligned with the solution proposed in this paper, work in this latter approach within the VITAL project (Petrolo et al., 2014) has not, to the best of our knowledge, been validated in an actual large-scale deployment. At the European level, the effort in coordinating the definition of a reference model for IoT platforms has been undertaken through the IoT Architectural Reference Model (ARM) developed within the IoT-A project (Bassi et al., 2013). IoT ARM only specifies the different views and models that should be followed when designing an actual IoT platform. Krco et al. (2014) summarize some of the IoT architectures that have been proposed using the IoT ARM. Kiljander et al. also take the IoT ARM as the reference and introduce semantic web enablement to promote interoperability in IoT scenarios. The common denominator for the solutions that follow this reference is the abstraction they create by exposing the physical world entities as virtual entities whose properties are fed via services, exposing the values captured by, or parameters acted upon, sensors and actuators respectively. This abstraction is also achieved by the IoT platform presented in this paper. Moreover, the actual mechanisms enabling this abstraction and handling of the information which is made available through the corresponding services is described. This latter aspect is however not detailed in the aforementioned works.

Last but not least, it is important to consider the efforts that are being made by the standardization bodies related to IoT. In this respect, there is some bias towards the Machine-to-Machine (M2M) domain in the current works. ETSI M2M TG[1] or the more recently created OneM2M (Swetina et al., 2014) have been working to standardize a common M2M service layer platform for globally applicable and access-independent M2M services. An exception to this is the specifications from OMA[2] in terms of Next Generation Service Interfaces (NGSI) for Context Management. Two standards OMA NGSI-9 and OMA NGSI-10 have provided the basis for the IoT Chapter specification in the

FIWARE[3] project. In any case, all the specifications are focused on the interfaces offering access to data as a service but ignore an important aspect, which as Daradkeh et al. (2012) already pointed out "is not covered yet in the specs: data persistence" (p.193) and data management.

4. IOT DATA MANAGEMENT PLATFORM

This section will describe in detail the insights of the different software components forming part of the SmartSantander platform for IoT Data Management, also known as the SmartSantander SEL platform (Service Experimentation Layer). The section is structured in three different levels: first of all, some of the underlying concepts behind the platform design are summarized; after that, the different building blocks in charge of gathering, managing and exposing the IoT data and what their mission as part of the platform is are presented; and finally, the various ways of interacting with the SmartSantander SEL platform are described.

4.1. SmartSantander SEL Underlying Concepts

4.1.1. Observations and Measurements Model

As has been previously shown, SmartSantander network is particularly heterogeneous in terms of devices. This means that information is generated in different devices, with different levels of computational power and usually managed by different parties. What is more, as new devices join the network, the complexity will grow and the governance and management of such a broad infrastructure must be kept under control.

In order to keep the interoperability simple, one of the first things to be modelled in the SmartSantander SEL platform was the data format. As the platform needs to be sufficiently independent to allow the inclusion of new sources of information (physical sensors or virtual data feeds) in the future, this data format was designed to be as generic as possible, able to deal with information from these new sources. The terminology followed implies that each *resource* generates *observations*, each of them including one or more *measurements*. As an example, the resource could be a programmable board including several sensors, identified by a Uniform Resource Name (URN), which generates an observation each minute including one measurement per sensor.

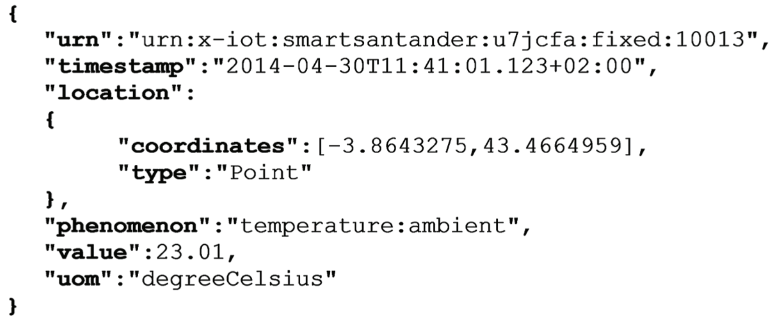
Observations in SmartSantander do not have any kind of associated state. They are always treated as independent chunks of information and once an observation enters the platform it cannot be modified in any way, or even deleted. As mentioned before, resources in SmartSantander always generate observations. However, it is often more convenient for those interested in the information generated not to consume that information as observations, but as single measurements.

An example of such an experiment could be a noise-sensor-based experiment, where information regarding luminosity sensors monitored by the same devices, and thus reported in the same observation, does not offer any extra value. As a result, SmartSantander manages two differentiated data models for measurements and observations, which can be seen in Figure 2 and Figure 3. The observation model essentially contains several related measurements, observed by the same resource at the same moment in time at the same location. As both models share the same fields, the impact on the usability is negligible. Anyway, the experimenter can always decide which data model to use, so there is no need to deal with different ones.

The observation (or measurement) data model includes:

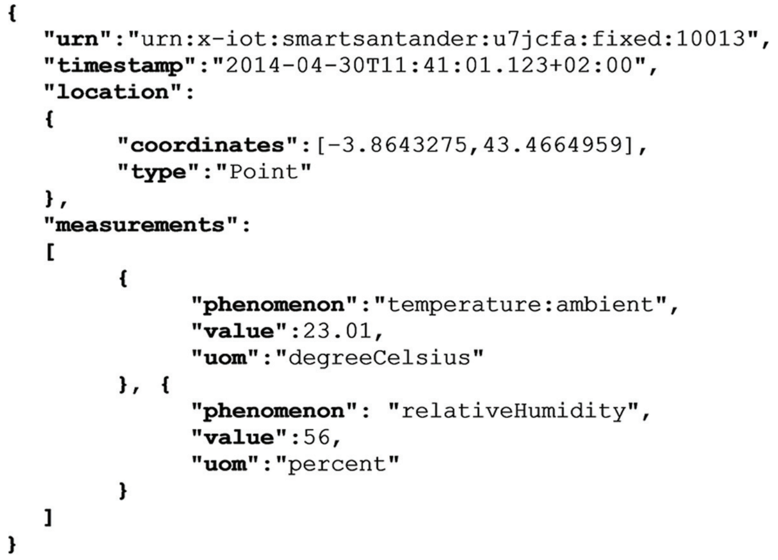
- **The resource identifier (URN):** It links the observation with the resource that generated the observation. Any observation coming from an unregistered resource is discarded. A URN can be split into several subdomains to ensure uniqueness;

Figure 2. SmartSantander single measurement example



```
{
   "urn":"urn:x-iot:smartsantander:u7jcfa:fixed:10013",
   "timestamp":"2014-04-30T11:41:01.123+02:00",
   "location":
   {
        "coordinates":[-3.8643275,43.4664959],
        "type":"Point"
   },
   "phenomenon":"temperature:ambient",
   "value":23.01,
   "uom":"degreeCelsius"
}
```

Figure 3. SmartSantander single observation example



```
{
   "urn":"urn:x-iot:smartsantander:u7jcfa:fixed:10013",
   "timestamp":"2014-04-30T11:41:01.123+02:00",
   "location":
   {
        "coordinates":[-3.8643275,43.4664959],
        "type":"Point"
   },
   "measurements":
   [
        {
             "phenomenon":"temperature:ambient",
             "value":23.01,
             "uom":"degreeCelsius"
        }, {
             "phenomenon": "relativeHumidity",
             "value":56,
             "uom":"percent"
        }
   ]
}
```

- **The observation timestamp:** It is usually generated by the device itself (or a provider's domain proxy). If not included, it is automatically generated once the observation enters the SmartSantander core platform;
- **The observation location:** It has to be formatted as a GeoJSON[4]. This way, resources not referring to a specific single point on the map (e.g. traffic sensors covering street sections) can also be included;
- **The specific measurement phenomenon:** A taxonomy of phenomena has been defined (SmartSantander phenomena) using well-known ontologies[5][6]7 to provide a unified dictionary to describe the sensor measurements' quantity kinds concept. New phenomena can be proposed for inclusion, but need to be accepted by an administrator prior to any new observation injection;
- **The specific unit of measurement (UoM):** A taxonomy of phenomena has been defined (SmartSantander UoM) using the same well-known ontologies. New units of measurement can be proposed for inclusion, but need to be accepted by an administrator prior to any new observation injection;

- **The specific measurement value:** This refers to the actual sensor information, but it can be a raw measurement coming from the sensor (e.g: millivolts) or a post-processed one, which is always defined by the UoM.

It is important to note that the information carried by an observation does not always provide the full picture. Extra sensor metadata, such as sensor precision, can be found in the resource description, which is relatively static and can be cached.

4.1.2. Subscriptions and Complex Queries

One of the big concerns for big IoT data platforms, and SmartSantander is not an exception, is how to offer to the data consumer a proper way to filter the available heterogeneous information to fit their requirements without heavily impacting usability. At the same time, these platforms must expose both historical and real-time data, and the way experimenters consume this information is not always the same and it may deeply impact the platform design decisions. Finally, it is necessary to establish a trade-off between simplicity and functionality so that requirements of skilled users of the platform are met while inexperienced users follow a gradual learning curve during their initial contact with the platform.

Having all that in mind, one of the first design decisions for the SmartSantander SEL platform was to provide a synchronous access layer on top of historical IoT data while, in parallel, offering an asynchronous system to avoid continuous polling for emulating real-time IoT data. In addition, another design decision was to lower the access barrier by offering neophyte users a basic RESTful API for accessing sensor information.

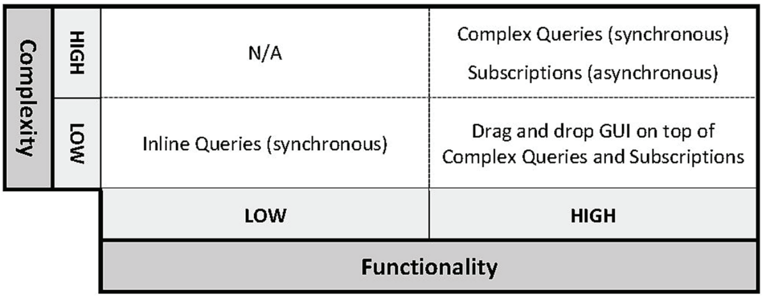
The approach adopted to maximise usability is to structure the access in the same way a user would face the problem, that is, first of all, *what* kind of information he or she wants to obtain, and then *where* and *when* that information was generated. Those three simple criteria are the basis of the whole IoT data access interface. This interface is simple enough for anyone to use, but at the same time, it covers the different user profiles SmartSantander has (in terms of technical background). A novel user can get an immediate overview of what can be achieved just by using a subset of the capabilities the synchronous system can offer through a web browser while trained users can benefit from the whole set of features to achieve fine-grained control of the kind of information they want to receive.

The mechanism used to control this advanced set of functionalities is known as *subscriptions* or *complex queries*, depending on whether we are referring to an asynchronous or synchronous system. A subscription allows a user to configure asynchronous notifications when a new observation or measurement is generated in the whole IoT infrastructure only if it matches the subscription definition. In contrast, when querying the synchronous system, complex queries can be used instead of URI parameters if the number of conditions grows. In this sense, SmartSantander's synchronous system defines two different kinds of queries: those using URI parameters, called *inline queries*; and those using complex queries, known as *referenced queries*. This provides a balance of complexity vs functionality that allows the SmartSantander SEL platform to be used with different user profiles. Figure 4 shows an overview of where all the aforementioned mechanisms are located in that balance.

In the RESTful terminology, both complex queries and subscriptions imply the creation of a persistent resource inside the SmartSantander platform. These resources can be managed by their owner and are formatted as JSON objects following the same *"what-where-when"* approach we have already introduced.

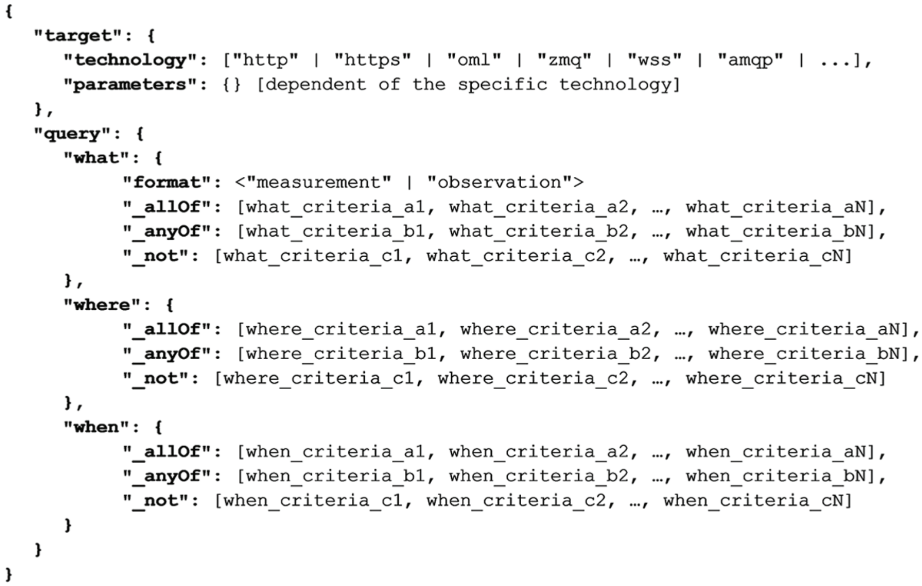
The generic definition of an asynchronous subscription is shown in Figure 5. The subscriptions model includes two main blocks: the "target", where the experimenter indicates the technology to be used for receiving the notifications; and the query, which defines what should trigger a notification. Complex Queries reuse the same format, but the nature of the synchronous system makes the "target" and "format" parameters unnecessary. As can be seen in Figure 5, the "query" attribute is split into three different categories, and different criteria can be combined to build them up. Combinations

Figure 4. SmartSantander platform complexity vs functionality



		Functionality	
		LOW	HIGH
Complexity	HIGH	N/A	Complex Queries (synchronous) Subscriptions (asynchronous)
	LOW	Inline Queries (synchronous)	Drag and drop GUI on top of Complex Queries and Subscriptions

Figure 5. SmartSantander asynchronous subscription format



```
{
   "target": {
      "technology": ["http" | "https" | "oml" | "zmq" | "wss" | "amqp" | ...],
      "parameters": {} [dependent of the specific technology]
   },
   "query": {
      "what": {
            "format": <"measurement" | "observation">
            "_allOf": [what_criteria_a1, what_criteria_a2, …, what_criteria_aN],
            "_anyOf": [what_criteria_b1, what_criteria_b2, …, what_criteria_bN],
            "_not": [what_criteria_c1, what_criteria_c2, …, what_criteria_cN]
      },
      "where": {
            "_allOf": [where_criteria_a1, where_criteria_a2, …, where_criteria_aN],
            "_anyOf": [where_criteria_b1, where_criteria_b2, …, where_criteria_bN],
            "_not": [where_criteria_c1, where_criteria_c2, …, where_criteria_cN]
      },
      "when": {
            "_allOf": [when_criteria_a1, when_criteria_a2, …, when_criteria_aN],
            "_anyOf": [when_criteria_b1, when_criteria_b2, …, when_criteria_bN],
            "_not": [when_criteria_c1, when_criteria_c2, …, when_criteria_cN]
      }
   }
}
```

can be made at different levels, so the granularity that can be achieved is very high. If no criteria are included in a subscription definition or a complex query, it is understood that the users want to access everything they are allowed to access.

Next, the modelling of the three different filtering criteria is described in detail.

4.1.3. The <what> Filtering Criterion

This criterion provides filtering capability for three different levels:

1. Only those measurements measuring phenomenon ‘P’ are relevant;
2. Among all measurements of phenomenon ‘P’, only those expressed with the unit of measurement ‘U’ are relevant;
3. Among all measurements of phenomenon ‘P’ with the unit of measurement ‘U’, only those with value ‘V’ are relevant.

Valid values of ‘P’ and ‘U’ are defined in the taxonomies created (SmartSantander phenomena, SmartSantander UoM). Moreover, different operations are supported to specify the desired valid values. Figure 6 shows the JSON model defining this filtering criterion. It is important to remark that the three levels are additive so, for example, it is not possible to specify a filter on the measurement value without including the UoM. Users are more likely to be interested in comparable measurements. Hence, extracting temperature measurements greater than 25 without knowing whether they are measured in Celsius or Kelvin does not have much sense.

4.1.4. The <where> Filtering Criterion

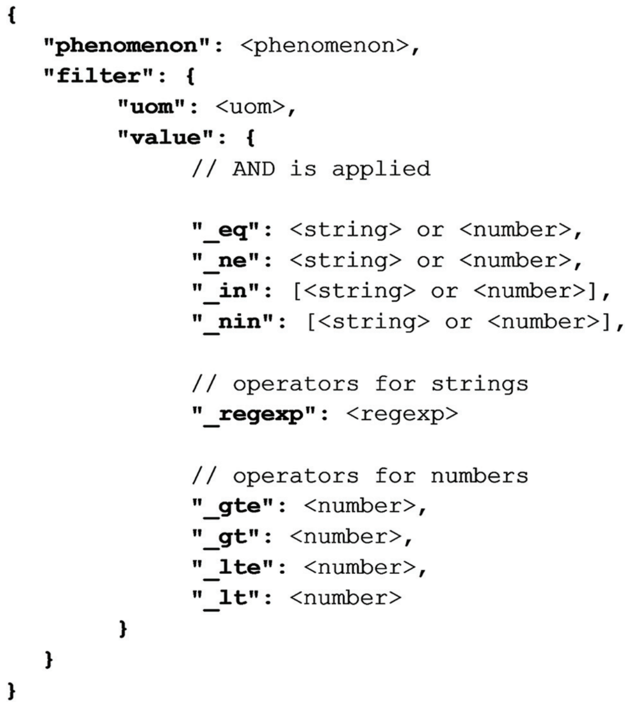
This filtering category allows a user to specify an observation/measurement source. This can be expressed in two ways: either as a set of resource URNs or as a geographical area described in GeoJSON format. Figure 7 shows the JSON model describing this criterion.

4.1.5. The <when> Filtering Criterion

This criterion is mostly used when accessing historical data, but can also be used to cover some special use cases.

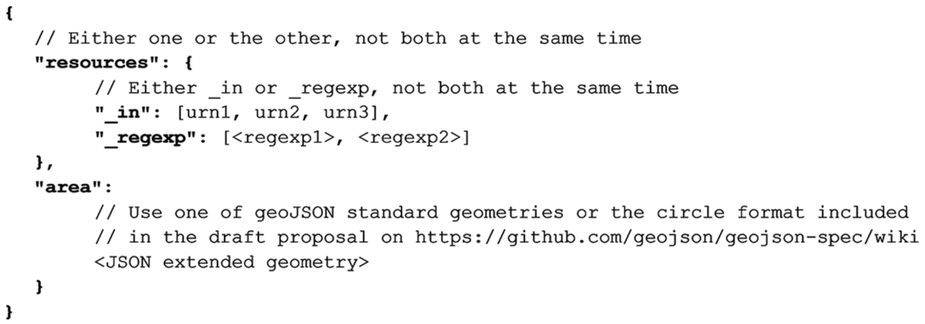
Figure 8 shows the structure of the “when” criterion. First of all, a “time” parameter can be used to define what kind of date should be used in the comparison. Valid values are UTC (which is the default value), local time (which is useful for example if a user wants to experiment with traffic congestion measurements in Santander and some other city in different time zones, as 10:00 UTC in Santander can be night time in other places), and insertion time (as observations are not always injected into the SmartSantander platform in real time). Then, time constraints can be specified in different ways:

Figure 6. SmartSantander subscription/complex query <what> criteria format



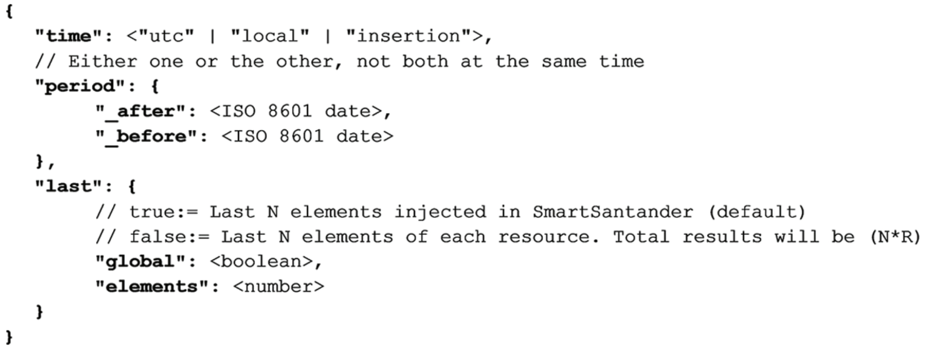
```
{
   "phenomenon": <phenomenon>,
   "filter": {
         "uom": <uom>,
         "value": {
               // AND is applied

               "_eq": <string> or <number>,
               "_ne": <string> or <number>,
               "_in": [<string> or <number>],
               "_nin": [<string> or <number>],

               // operators for strings
               "_regexp": <regexp>

               // operators for numbers
               "_gte": <number>,
               "_gt": <number>,
               "_lte": <number>,
               "_lt": <number>
         }
   }
}
```

Figure 7. SmartSantander subscription/complex query <where> criteria format



```
{
   // Either one or the other, not both at the same time
   "resources": {
        // Either _in or _regexp, not both at the same time
        "_in": [urn1, urn2, urn3],
        "_regexp": [<regexp1>, <regexp2>]
   },
   "area":
        // Use one of geoJSON standard geometries or the circle format included
        // in the draft proposal on https://github.com/geojson/geojson-spec/wiki
        <JSON extended geometry>
   }
}
```

Figure 8. SmartSantander subscription/complex query <when> criteria format



```
{
   "time": <"utc" | "local" | "insertion">,
   // Either one or the other, not both at the same time
   "period": {
        "_after": <ISO 8601 date>,
        "_before": <ISO 8601 date>
   },
   "last": {
        // true:= Last N elements injected in SmartSantander (default)
        // false:= Last N elements of each resource. Total results will be (N*R)
        "global": <boolean>,
        "elements": <number>
   }
}
```

- Specific ISO8601 dates can be used to limit the results. This is the most usual way of filtering historical data when dealing with the synchronous subsystem;
- ISO8601 durations, which are referenced to the moment the request is performed. As an example, this can be used if we want to retrieve information from the last hour;
- A fixed quantity, which specifies how many of the last observations (or measurements) a user wants to retrieve.

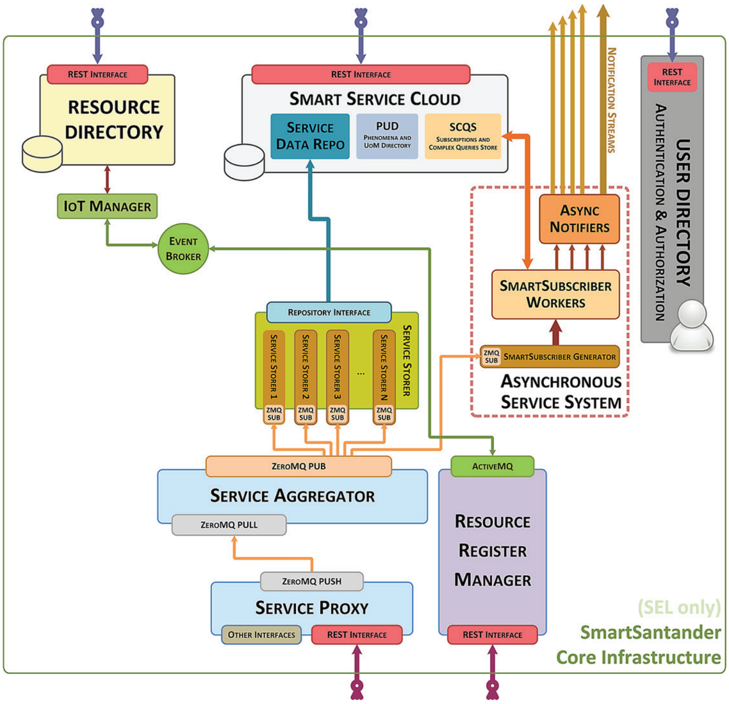
As stated before, due to the real-time nature of asynchronous subscriptions, applying this kind of filtering criteria does not seem to have sense. However, there is a use case when this can be interesting: as observations can be inserted in a deterministic way with timestamps in the past, or the provider can just be using incorrect timestamps, being able to subscribe to real-time injected observations that are dated in the past (using the durations approach) can be useful.

4.2. SmartSantander Data Management System Specification

Figure 9 depicts the functional architecture of the complete SmartSantander SEL platform. As this paper is focused on how SmartSantander manages sensor data, components related to resource description or authentication and authorization are considered beyond its scope. Nevertheless, a brief summary of the role of each component is included here to help the reader to understand how the whole platform behaves.

Several subsystems can be differentiated within the SEL platform:

Figure 9. SmartSantander SEL functional architecture



- The Security sub-system is responsible for the authentication, authorization, and management of the users in the system. Its main component is the User Directory;
- The Resource sub-system handles resource description information. Its main components are the Resource Register Manager and the Resource Directory which is the actual repository for the descriptions of the infrastructure resources. There is another component, known as IoT Manager, which enables interoperability between different domains;
- The mission of the Taxonomy sub-system, essentially composed of the Phenomena and UoM Directory (PUD), is the definition of the accepted vocabulary associated with the different sensor parameters supported;
- The Information sub-system is in charge of adapting and storing observations to be used both in the synchronous and asynchronous service system. As the focus of this article is to dig into this sub-system, the next subsections include detailed aspects of its different components.

All these sub-systems are provided to the outside world through a RESTful API which will be described in the next section.

It is interesting to remark that, even though RESTful APIs follow a request-response communication scheme, in the context of asynchronous services, system experimenters can use such an interface to synchronously manage their own subscriptions. Notifications are then delivered to them out of band according to the *target* information contained in those subscriptions.

4.2.1. Service Proxy

This is the entry point for any infrastructure provider wanting to insert any kind of IoT data in the SmartSantander SEL platform.

Newly injected information is exposed as part of the IoT API, and this can be done by issuing a POST request to a specific endpoint including one or more observations, see Figure 3 in the message body. The Service Proxy will filter out any observation coming from an unregistered resource or including a capability (the combination of a phenomenon plus a UoM) not listed in its resource description.

This component is also in charge of adding any extra information not included in the received observation, such as the location information if the source resource is registered as a fixed one, or the timestamp if it is not included by the infrastructure provider.

In addition, the Service Proxy will generate the topic list which will be used later on to publish those resource observations (see Service Aggregator and Service Storer description). Topics are used to enable quick filtering per application domain and federation.

In order to enhance scalability and performance, this component is developed following an event-driven concurrency model using Node.JS technology.

For deployment, the Service Proxy can be horizontally scaled just by replicating it on different machines and properly configuring the load balancer. It uses a ZeroMQ[8] based push/pull architecture to send the observations to the next component in the pipeline, which is the Service Aggregator.

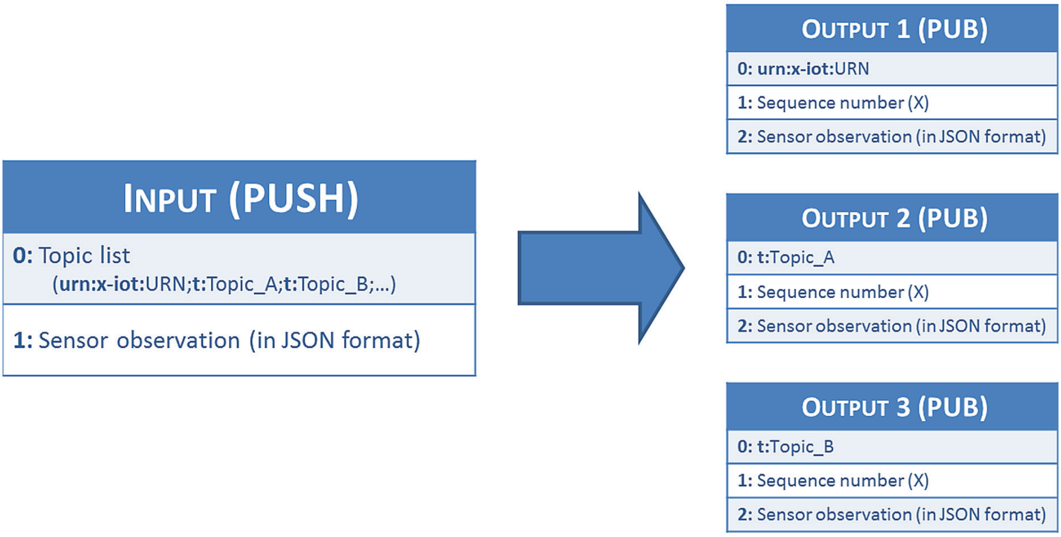
4.2.2. Service Aggregator

This component serves as a broker for all the observations gathered. Different Service Proxy instances push observations towards the Service Aggregator, which unfolds the messages and publishes them on the different specified topics using a ZeroMQ based pub/sub mechanism. This mechanism can only be used by internal SmartSantander components, and it is not publicly available outside the SmartSantander core system.

As this is a central component of the architecture, its performance is critical. For this reason, it has been implemented using native C and low-level ZeroMQ libraries. Its design is based on an event loop approach.

The Service Aggregator can be considered as a routing component, and does not process JSON observations at all. It also includes a sequence number to detect potential duplication of consumers subscribed to multiple topics. Its functionality can be summarized in Figure 10. In essence, this

Figure 10. Service aggregator overview



component can be considered as a stateless message switch in the middle of a star topology, which acts as an intermediary between observation producers and consumers.

4.2.3. Service Storer

Federation is enabled through this component. Each of the different modules that are part of it is in charge of transforming and redirecting relevant observations to platforms with which SmartSantander is federated. Only the module in charge of storing observations on the SmartSantander Service Data Repository (so-called SDR Storer) is compulsory. The FIWare IDAS/DCA platform is an example of an external repository that can receive information from a Service Storer module.

Examples of some of the data models that those modules handle are UltraLight or JSON-LD (for semantically annotated data).

4.2.4. Service Data Repository

This component is the one supporting persistence for all the service information generated inside SmartSantander.

The current implementation is based on NoSQL solutions, in particular on a 3rd party distribution of MongoDB called TokuMX[9]. The main advantage this distribution offers to SmartSantander is the possibility of using time-series partitioned collections. Thus, data in SmartSantander is virtually split into different collections based on the observation timestamp and the search efficiency grows. In addition, this also enables the export of a single partition from the primary database, moving old historic observations to secondary storage solutions to avoid infinite growth of disk space requirements on the primary server and to improve efficiency. Sharing logic between these two platforms is handled at application level. At the moment, only two different storage levels are envisioned, but extra levels can be added in the future if needed.

The SDR Storer module is continuously writing to the Service Data Repository, hence, TokuMX document level locking characteristic helps to improve the performance of concurrent read/write workloads. As observations are never updated, documents are created without padding and fragmentation is not a problem. This, together with disk compression, helps to reduce the amount of disk space needed.

On the other hand, read access is controlled by the SmartSantander IoT API. In order to speed up these query operations, different compound indexes have been created based on common user interaction patterns.

Finally, high availability is achieved by replication, although user queries always reach the primary server, which is hosted using an array of Solid State Disk (SSD) disks in a RAID 10 configuration.

4.2.5. Complex Queries Store

This component is the repository keeping the complex queries made by the users. It is part of the Subscriptions and Complex Queries Store (SCQS). Read and Write operations are performed through the SmartSantander IoT API. It can act as a cache, for improving usability, as well as for reducing response times to similar queries.

From a deployment perspective, it is based on an official MongoDB database deployed on a high availability cluster in a similar way to the Service Data repository. Sharding is not yet a requirement due to the workload it supports, but a username can be used as a sharding key if the demand grows.

4.2.6. SmartSubscriber Generator

The role of this component is to handle all the observations from the Service Aggregator in a similar way to how SDR Storer does. However, instead of storing it for persistence, it generates background tasks to be executed by the different SmartSubscriber Workers. These tasks are stored using Redis Queues[10] (Macedo, 2011).

4.2.7. Subscription Store

This component keeps the information about subscriptions registered inside the platform. It is part of the Subscriptions and Complex Queries Store (SCQS). Read and Write operations are performed through the SmartSantander IoT API. Active subscriptions are cached in the User Subscriptions Cache (USC) to be consumed by the SmartSubscriber Workers. Internally, it has the same development as the complex queries store.

4.2.8. SmartSubscriber Workers

These workers are in charge of evaluating all the active subscriptions in the USC for a single observation. In this sense, after applying authorization rules, SmartSubscriber workers push the resulting notifications in the different notification channels. There is one different notification channel for each supported notification technology such as RestHooks (Guinard et al., 2011), OML (Singh et al., 2005) or websockets (Lubbers and Greco, 2010). Both USC and Notification channels are based on a Redis[11] deployment.

4.2.9. Async Notifiers

Async Notifiers are the modules emitting the notifications to the end users based on the subscription target. Each notifier handles a different technology, and they can be horizontally scaled by replication if needed. Async notifiers can use a loopback mechanism to inform the system if a notification fails because the other endpoint is not listening and then disable the subscription.

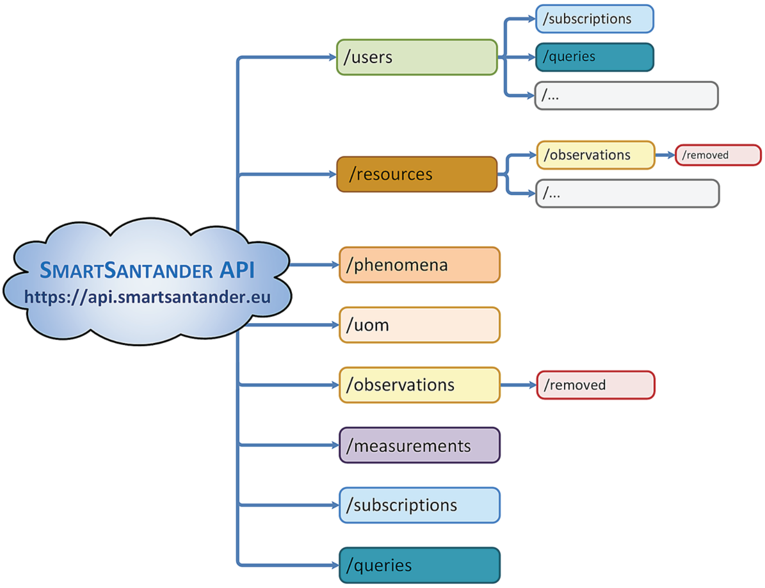
Due to the necessary level of concurrency these components have, the implementation has been carried out using Python asyncio coroutines. This programming model can handle a lot of parallel tasks while keeping the resource consumption within bounds.

4.3. SmartSantander SEL IoT API

As has been mentioned throughout the paper, access to the SmartSantander SEL platform functionalities is enabled through a REST interface which gives access to the different resources/devices, including characteristics and available data (i.e. historical and last values).

Each of the REST resources mapped in the tree representation shown in Figure 11 enables access to the different SmartSantander testbed functionalities.

Figure 11. SmartSantander IoT API resource tree



The organization of the API allows multiple ways of accessing the platform functionalities depending on the necessities and focus from the experimenters. In terms of IoT data access, the */subscriptions*, and */queries* resources enable access to the above-described subscriptions and complex queries respectively. As has also been mentioned when describing the Service Proxy module, the same API is used for inserting data coming from the IoT infrastructure through the */observations* resource. Other API resources deal with other platform functionalities which are beyond the scope of this paper.

5. CONCLUSION

This paper has provided an ample overview of the IoT data management platform implemented within the SmartSantander project to make available the data generated by a large-scale IoT deployment composed of thousands of sensors.

In addition to the description of the developed system, the main requirements and design considerations adopted for the platform have also been summarized. It is important to highlight that these considerations have been derived and fine-tuned as a result of the experience gathered throughout the actual roll-out of the IoT infrastructure as well as from the lessons learnt during insightful rounds of external users consuming information generated in the smart city deployment and challenging extensions with additional sources of information. Thus, the resulting detailed specification comes after fine tuning enabled by day by day management of the infrastructure, support of external experimentation and smart city service creation. The information and query models created have been developed in an integrated manner so that information consumption can be handled efficiently and flexibly, supporting complex queries and subscriptions.

Platform specification and implementation are instances for which development decisions have been made, but the aforementioned requirements establish a canonical base for systems handling large amounts of IoT data which is supported by real-world experience.

The development presented in this paper has its roots in the observation and measurement models that were defined in the SmartSantander project. These models were not aimed at supporting interoperability of the developed platform with other IoT deployments that have to adapt to these models before the data that they generate can be accessed through the interfaces described in this paper. This is an important limitation that is being addressed by integrating semantic web solutions within the SmartSantander platform so that it is possible to tap into the great potential that the exchange of semantic information can offer for the development of future applications for the increasing number of smart devices in circulation (Diaz, Sigüenza, Bernat, & Hernández, 2014). Currently, work is being done applying ontology-based modelling so that concepts, instances, and relationships can formally and comprehensively represent the knowledge and foster interoperability, formality and reusability (Li, Eckert, Martinez, & Rubio, 2015) of the platform.

REFERENCES

Guinard, D., Trifa, V., Mattern, F., & Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In *Architecting the Internet of Things* (pp. 97–129). Springer Berlin Heidelberg. doi:10.1007/978-3-642-19157-2_5

Hancke, G. P., Silva, B., & Hancke, G. Jr. (2013). The role of advanced sensing in smart cities. *Sensors (Basel, Switzerland)*, *13*(1), 393–425. doi:10.3390/s130100393 PMID:23271603

Kiljander, J., D’Elia, A., Morandi, F., Hyttinen, P., Takalo-Mattila, J., Ylisaukko-Oja, A., & Cinotti, T. S. et al. (2014). Semantic interoperability architecture for pervasive computing and Internet of Things. *Access, 2*, 856–873. doi:10.1109/ACCESS.2014.2347992

Kitchin, R. (2014). The real-time city? Big data and smart urbanism. *GeoJournal*, *79*(1), 1–14. doi:10.1007/s10708-013-9516-8

Kourtit, K., Nijkamp, P., & Arribas-Bel, D. (2012). Smart cities perspective – A comparative European study by means of self-organizing maps. *Innovation (Abingdon)*, *25*(2), 229–246. doi:10.1080/13511610.2012.660330

Krco, S., Pokric, B., & Carrez, F. (2014, March). Designing IoT architecture (s): A European perspective. *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 79-84). IEEE.

Lanza, J., Sánchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., & Gutiérrez, V. (2015). Large-Scale Mobile Sensing Enabled Internet-of-Things Testbed for Smart City Services. *International Journal of Distributed Sensor Networks*, *2015*, 1–15. doi:10.1155/2015/785061

Li, X., Eckert, M., Martinez, J. F., & Rubio, G. (2015). Context Aware Middleware Architectures: Survey and Challenges. *Sensors (Basel, Switzerland)*, *15*(8), 20570–20607. doi:10.3390/s150820570 PMID:26307988

Lubbers, P., & Greco, F. (2010). *HTML5 web sockets: A quantum leap in scalability for the web*. SOA World Magazine.

Macedo, T., & Oliveira, F. (2011). *Redis cookbook*. O’Reilly Media, Inc.

Mitton, N., Papavassiliou, S., Puliafito, A., & Trivedi, K. S. (2012). Combining Cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, (1), 1–10.

Nam, T., & Pardo, T. A. (2011, June). Conceptualizing smart city with dimensions of technology, people, and institutions.*Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times* (pp. 282-291). ACM. doi:10.1145/2037556.2037602

Nordström, E., Gunningberg, P., Rohner, C., & Wibling, O. (2007, June). Evaluating wireless multi-hop networks using a combination of simulation, emulation, and real world experiments.*Proceedings of the 1st international workshop on System evaluation for mobile platforms* (pp. 29-34). ACM. doi:10.1145/1247721.1247728

Ojala, T., Kukka, H., Lindén, T., Heikkinen, T., Jurmu, M., Hosio, S., & Kruger, F. (2010, May). UBI-hotspot 1.0: Large-scale long-term deployment of interactive public displays in a city center.*Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services (ICIW)* (pp. 285-294). IEEE. doi:10.1109/ICIW.2010.49

Panasonic. (n. d.). Fujisawa Sustainable Smart Town. Retrieved from http://panasonic.net/es/fujisawasst/

Petrolo, R., Loscri, V., & Mitton, N. (2014). Towards a Cloud of Things smart city. *IEEE COMSOC MMTC E-Letter, 9*(5).

Pulso de la Ciudad. (n. d.). Sensado Participativo. Retrieved from http://www.elpulsodelaciudad.com/

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., & Pfisterer, D. et al. (2014). SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, *61*, 217–238. doi:10.1016/j.bjp.2013.12.020

Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. In J. Domingue, et al. (Eds.), Future Internet Assembly (pp. 431-446). Springer.

Singh, M., Ott, M., Seskar, I., & Kamat, P. (2005, February). ORBIT Measurements framework and library (OML): motivations, implementation and features. *Proceedings of the First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities Tridentcom '05*. (pp. 146-152). IEEE.

SmartSantander API Documentation. (n. d.). Retrieved from https://api.smartsantander.eu/docs

SmartSantander phenomena taxonomy. (n. d.). SmartSantander. Retrieved from https://api.smartsantander.eu/phenomena

SmartSantander Project FP7-ICT-2009-5-257992. (2010). SmartSantander. Retrieved from http://www.smartsantander.eu

SmartSantander units of measurement (uom) taxonomy. (n. d.). SmartSantander. Retrieved from https://api.smartsantander.eu/uom

Strickland, E. (2011). Cisco bets on South Korean smart city. *Spectrum, IEEE, 48*(8), 11-12.

Swetina, J., Lu, G., Jacobs, P., Ennesser, F., & Song, J. (2014). Toward a standardized common m2m service layer platform: Introduction to onem2m. *Wireless Communications, 21*(3), 20–26. doi:10.1109/MWC.2014.6845045

(2010, April01). Urbanization and health. *Bulletin of the World Health Organization*, *88*(4), 245–246. doi:10.2471/BLT.10.010410

Vilajosana, I., Llosa, J., Martinez, B., Domingo-Prieto, M., Angles, A., & Vilajosana, X. (2013). Bootstrapping smart cities through a self-sustainable model based on big data flows. *Communications Magazine, 51*(6), 128–134. doi:10.1109/MCOM.2013.6525605

Zygiaris, S. (2013). Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems. *Journal of the Knowledge Economy*, *4*(2), 217–231. doi:10.1007/s13132-012-0089-4

ENDNOTES

[1] ETSI Machine-to-Machine Communications. http://www.etsi.org/website/technologies/m2m.aspx

[2] Open Mobile Alliance. http://openmobilealliance.org/

[3] FIWARE Project. https://www.fiware.org/

[4] GeoJSON http://geojson.org/

[5] Quantities, Units, Dimensions and Types (QUDT) Ontology http://www.qudt.org/

[6] Quantities and Units Ontology derived from UNECE / CEFACT Rec. 20 and Gellish (qu-rec20) http://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu-rec20.html

[7] Semantic Web for Earth and Environment Terminology (SWEET) Ontology https://sweet.jpl.nasa.gov

[8] Distributed messaging – zeromq. http://zeromq.org/

[9] https://www.percona.com/doc/percona-tokumx

[10] http://python-rq.org/

[11] http://redis.io/

Jorge Lanza received a degree in telecommunications engineering from University of Cantabria (UC), Spain, in 2000 and the PhD in 2014. Since then he has been a researcher at the Network Planning and Mobile Communications Laboratory of that university, where he collaborates giving lectures and practical courses at the university and in national and international conferences. He has participated in several research projects, national and international, with both private and public funding. As a member of the Technological Observatory for Smart Cards at the University of Cantabria (OTTIUC), he also works with highly regarded manufacturers, banking entities and mobile operators acquiring experience in smartcard technology. The work carried out in the OTTIUC includes the design and deployment of the multi-application smartcard technology and environment provided by Santander bank and used in up to 300 universities around the world. His current research in combined mobility and security for the wireless Internet has been carried out based on the merging of wireless technologies and smartcards for current and next generation networks. In this work, it plays a key role in the new emerging NFC technology (he has been working on it since 2005) now being included in most mobile phones, enabling new scenarios for personal communications and relationships with the environment, for context retrieval, identification, authentication, etc.

Pablo Sotres works as research fellow in the Network Planning and Mobile Communications Laboratory, which belongs to the Communications Engineering department at the University of Cantabria, Spain. He received Telecommunications Engineering degree from the University of Cantabria in 2008. He is currently involved in different European projects framed under the smart city paradigm, such as SmartSantander; and related to inter-testbed federation, such as Fed4FIRE.

Luis Sanchez received both the Telecommunications Engineering and PhD degree from the University of Cantabria, Spain, in 2002 and 2009 respectively. He is Associate Professor at the Department of Communications Engineering at the University of Cantabria. He is active on IoT-enabled smart cities, meshed networking on heterogeneous wireless scenarios and optimization of network performance through cognitive networking techniques. He has a long research record working on projects belonging to the 5th, 6th, 7th and H2020 EU Framework Programmes. He has authored more than 60 papers at international journals and conferences and co-authored several books.

Jose Antonio Galache received both the Telecommunications Engineering and PhD degree from the University of Cantabria, Spain, in 2004 and 2013 respectively. He is currently active on different European projects framed under the smart city paradigm, such as SmartSantander, as well as others fostering collaboration between Europe and Japan, such as ClouT project, sharing experiences within smart cities framework. His research interests are on WSNs, wireless routing protocols and middleware platforms for managing heterogeneous networks, as well as M2M communications within the frame of a smart city.

Juan Ramón Santana is a Telecommunication Engineer graduated in 2010 from the University of Cantabria. He is currently working as research fellow in the Network Planning and Mobile Communications Laboratory, a telecommunication research group from the same university. Prior to his current occupation, he did an internship at the University of Strathclyde (Glasgow) working on IoT solutions. He has been involved in several projects, such as SmartSantander, EAR-IT or FESTIVAL, and European collaborative projects related to the Smart City paradigm and the Internet of Things. Among his research interests are WSN (Wireless Sensor Networks), M2M communications and Mobile phone application research.

Verónica Gutiérrez received the Telecommunications Engineering degree from the University of Cantabria, Spain, in 2000. She is currently a senior researcher at the Network Planning and Mobile Communications Laboratory at the University of Cantabria. She is active on different European projects framed under the smart city paradigm, such as SmartSantander, as well as others like RADICAL and Organicity, leveraging the Living Lab concept, sharing experiences within smart cities framework. Her research interests are on participatory sensing, augmented reality and middleware platforms for IoT-based smart city services development. She is currently working towards her PhD.

Luis Muñoz received both the Telecommunications Engineering degree and a PhD from the Polytechnical University of Cataluña (UPC), Spain, in 1990 and 1995, respectively. He is head of the Network Planning and Mobile Communications Laboratory belonging to the Communications Engineering Department (DICOM) at the University of Cantabria, Spain. His research focuses on advanced data transmission techniques, heterogeneous wireless multihop networks and applied mathematical methods for telecommunications. He has participated in several National and European research projects belonging to the 4th, 5th, 6th and 7th Framework Program in which he is technical manager of SmartSantander. He has published over 150 journal and conference papers. He serves as editor of several journals. In parallel to this activity, he serves as consultant for the Spanish Government as well as for different companies in Europe and USA.

Call for Articles

International Journal on Semantic Web and Information Systems

Volume 12 • Issue 4 • October-December 2016 • ISSN: 1552-6283 • eISSN: 1552-6291

An official publication of the Information Resources Management Association

MISSION

The **International Journal on Semantic Web and Information Systems (IJSWIS)** is an archival journal that publishes high quality original manuscripts in all aspects of Semantic Web that are relevant to computer science and information systems communities. IJSWIS is an open forum aiming to cultivate the Semantic Web vision within the information systems research community. The main focus is on information systems discipline and working towards the delivery of the main implications that the Semantic Web brings to information systems and the information/knowledge society.

COVERAGE AND MAJOR TOPICS

The topics of interest in this journal include, but are not limited to:

Beyond Semantic Web (e.g., extending meaning with perception and experience) • Enterprise application integration • From e-government to e-democracy • Integration with other disciplines • Intelligent Systems • Metadata-driven (bottom-up) versus ontology-driven (top-down) SW development • New Semantic Web enabled business models • New Semantic Web enabled information systems • New Semantic Web enabled tools for the citizen/ learner/ organization/ business • Ontologies, folksonomies, and associated knowledge representation issues • Real world applications towards the development of the Knowledge society • Semantic enabled business intelligence • Semantic Web applications on the Web, enterprises, desktops, personal and mobile devices, e-science and e-government applications, and associated issues of provenance, trust, privacy, security, quality, scalability, and performance • Semantic Web data management • Semantic Web issues, challenges, and implications in each of the IS research streams • Semantics and human computer interfaces including visualization and mashups • Semantics in business processes and distributed computing and services • Social Semantic Web and people Web • Standards

ALL INQUIRIES REGARDING IJSWIS SHOULD BE DIRECTED TO THE ATTENTION OF:

Miltiadis D. Lytras, Editor-in-Chief • IJSWIS@igi-global.com

ALL MANUSCRIPT SUBMISSIONS TO IJSWIS SHOULD BE SENT THROUGH THE ONLINE SUBMISSION SYSTEM:

http://www.igi-global.com/authorseditors/titlesubmission/newproject.aspx

IDEAS FOR SPECIAL THEME ISSUES MAY BE SUBMITTED TO THE EDITOR(S)-IN-CHIEF

PLEASE RECOMMEND THIS PUBLICATION TO YOUR LIBRARIAN

For a convenient easy-to-use library recommendation form, please visit:
http://www.igi-global.com/IJSWIS