Citizen-Centric Data Services for Smarter Cities

Unai Aguilera[1, 2], Oscar Peña[1, 2], Oscar Belmonte[2], Diego López-de-Ipiña[1, 2]

[1] DeustoTech - Fundación Deusto, Avda Universidades, 24, 48007, Bilbao, Spain. e-mails: unai.aguilera@deusto.es, oscar.pena@deusto.es, dipina@deusto.es

[2] Facultad Ingeniería, Universidad de Deusto, Avda. Universidades, 24, 48007, Bilbao, Spain.

[3] Department of Computer Languages and Systems, Universitat Jaume I, 12071 Castellon, Spain. e-mail: oscar.belmonte@lsi.uji.es

Abstract

Smart Cities use Information and Communication Technologies (ICT) to manage more efficiently the resources and services offered by a city and to make them more approachable to all its stakeholders (citizens, companies and public administration). In contrast to the view of big corporations promoting holistic “smart city in a box” solutions, this work proposes that smarter cities can be achieved by combining already available infrastructure, i.e., Open Government Data and sensor networks deployed in cities, with the citizens’ active contributions towards city knowledge by means of their smartphones and the apps executed in them. In addition, this work introduces the main characteristics of the IES Cities platform, whose goal is to ease the generation of citizen-centric apps that exploit urban data in different domains. The proposed vision is achieved by providing a common access mechanism to the heterogeneous data sources offered by the city, which reduces the complexity of accessing the city’s data whilst bringing citizens closely to a *prosumer* (double consumer and producer) role and allowing to integrate legacy data into the cities’ data ecosystem.

Keywords: Smarter cities; Data Services; Citizen-centric Apps; Open Data; Legacy Data

1. Introduction

Smart Cities aim to improve the service they deliver to their citizens both in terms of economic (bigger efficiency) and social (effectiveness addressing their stakeholders’ needs and desires) impact. The achievement of these goals depends not only on data provided by governments, authorities and decision-makers; but also on data provided by citizens, as they can be seen as intelligent sensors [1]. All data gathered, either automatically by the city sensors or by contributing citizens, who act as mobile sensors in their daily interactions with the city, is useful to take earlier and better decisions, providing improved services to the citizens. Such combined knowledge is usable, for example, to intervene in the cities’ infrastructures maintenance or traffic management.

The IES Cities project[1] focuses on promoting user-centric mobile services that exploit open data combined with user-supplied data. The distinguishing feature of the project is the contribution with a platform that enables the creation of data centric apps, which also copes with the active data contributions coming from citizens. This platform is the result of a European project (Competitiveness and Innovation Framework Programme) evaluated in four different European cities: Zaragoza and Majadahonda (Spain), Bristol (United Kingdom), and Rovereto (Italy), allowing citizens to obtain the most of the data offered by the city and,

[1] www.iescities.eu

especially to application developers, to contribute with the creation of new urban applications. The platform allows citizens to create and consume mobile web-compatible apps, which exploit Open Government and user-generated data related to their cities.

Something especially remarkable about the IES Cities approach is that, to the best of our knowledge, there is no other project that has considered before so much the extent of the impact that the users may have on improving, extending and enriching the open data which city services are usually based on. Citizens are the main stakeholders of our envisaged smart city-enabling platform through an urban services' ecosystem. Still, such novel public service apps are also of interest for both private companies and public administrations of the city, which might consume, provide and even exploit such apps.

The IES Cities platform is inspired by the philosophy that lies behind the "Apps for Smart Cities Manifesto" [7], which confronts the "smart city in a box" [8] solutions offered by big corporations such as IBM, Cisco or Siemens. The main statement behind this manifesto is the following: "The city must become like the Internet, i.e., enabling creative development and easy deployment of applications which aim to empower the citizen". It highlights five "technologies that matter" for cities in 2020: mobile broadband, smart personal devices, government-sponsored cloud and open-source public databases to promote grassroots innovation and "public interfaces".

Particularly, this work addresses three main scientific challenges towards contributing to more citizen inclusive cities, i.e., Smarter Cities:

a) To adapt heterogeneous structured and non-structured data from council repositories, web sites and social networks.
b) To facilitate the creation of urban apps by end developers.
c) To serve as an easy-to-deploy starting point for those city councils and administrations that want to adopt open data and open services approach as enablers of Smart Cities.

The IES Cities platform addresses the first challenge by providing a uniform access mechanism for developers that eases the usage of city's heterogeneous data, i.e., broad data[2] associated to a city, which is an open research topic in Smart City's data infrastructures [2]. IES Cities enables app developers to use and contribute to the city's repositories in a homogeneous way. On the other hand, combining a simplified access to heterogeneous data sources with the metadata management potential provided by the platform, the IES Cities project provides app developers with a set of tools that eases creation of urban applications, aimed to solve the second of the previously introduced challenges. Finally, the platform solves the third of the previous challenges thanks to its architecture and the mechanisms that it provides to improve the data management cycle and the integration of already existing data sources with those newly created by the city during its migration to the Smart City vision.

The structure of the paper is the following. Section 2 introduces some background work and focuses on the differences of our approach compared with other solutions providing citizen data services for Smart Cities. On the other hand, Section 3 presents the characteristics of the IES Cities platform and its main components, whereas Section 4 summarises some of the functionalities provided by the *Query Mapper* module, whose goal is to facilitate the uniform exploitation of heterogeneous urban datasets. Section 5 explains how the data management and how data services are discovered and used with the proposed platform. Section 6 summarizes the validation of the system from three different views: city councils participation, third-party developers and, in addition, the creation of two exemplary applications. Finally, Section 7 summarizes this paper's contributions and outlines future work.

[2] http://www.ibmbigdatahub.com/blog/not-big-data-broad-data

2. Background and related work

This work presents the IES Cities platform, which promotes the creation of user-centric mobile services constructed around the Internet of Things (IoT) [9], Linked and Open Data [12] and Crowd-sourced data [13]. The development of embedded devices with wireless connection to the Internet and with capability to run a web server has led to research efforts on Internet protocols not only to access information but also to republish, discover and compose web services based on the so called Internet of Things (IoT).

The integration of IoT technologies into Smart Cities allows citizens to obtain knowledge about their surroundings and to contribute with new data captured through their mobile devices. In addition, the data can be accessible as data sources queriable and contributable through common access mechanisms (i.e., JSON), which allows to provide a homogenous access to the different data providers available in the city.

During the past years, wireless sensor networks have received special attention, as the devices connected to these networks provide time-continuous sensing information. Human-Centric Wireless Sensor Networks (HWSN) [10, 11] are a particular case of opportunistic Sensor Networks that are composed of three types of nodes: regular sensors, information holders (keep alive the information delivered by the sensors), and human-based sensors. HCSN shares some commonalities with the IES Cities approach, although there are remarkable differences between both approaches. For example, citizens produce the data shared in the IES Cities, but also public administration and third party companies participate by providing information, which in turn implies some pre-processes of the information before being available in the IES Cities. However, our proposal can integrate information provided by this kind of networks. The network can be connected as an external dataset accessible through the Query Mapper module explained in Section 4 and, therefore, queriable through a common query language (i.e., SQL).

On the other hand, some works in the IoT field [16, 17, 20, 23] suggest that REST is the common architectural design to facilitate a smooth integration of sensors in the Internet. IES Cities leverages on these service access and data serialization technologies to provide its vision by providing an access to different heterogeneous data sources.

Since the introduction of the term crowdsourcing and crowd-sourced data [24], there is a tendency to the construction of collaborative platforms where users contribute by collecting different types of data: encyclopaedic data (e.g., Wikipedia [3]), cartography data (e.g., OpenStreetMaps [4]) or traffic data (e.g., Waze [32]), among other examples. The resulting datasets are useful not only for their primary purposes, but also for other applications: education, urban planning, disaster relief, etc.

In the Smart City context, there are some projects based on crowd sourcing and collaboration to give citizens the chance to participate providing data about the city (e.g., FixMyStreet [5] or FixMyTransport [6]). This way, citizens can be viewed as mobile sensors that monitor the variables of the city, and the data provided by them as crowd-sourced data. Data commonly provided by citizens are atmospheric (temperature, pressure, air humidity, soil humidity, etc.) or environmental data (gas emissions, pollutants, noise level, etc.) which is measured by the physical sensors of their smartphones or other connected sensing devices. Furthermore, this type of applications also allow citizens to directly sense and edit the data to include, for example, information about damaged street furniture, bumps in the road, vandalism, etc.

Based on the previous ideas, the IES Cities project provides a set of tools that eases the access and contribution to existing datasets by offering a homogenous access to the data and other utilities to the construction of data centric applications. The IES Cities project bases on some existing proposals where the final user (i.e., citizens in the case of IES Cities) are a fundamental part of the development process [14, 15].

In this case, the platform is designed to be flexible enough to connect data sources from different domains and formats, accessing the data through a RESTful API. The idea of transforming datasets to access them through a REST API already was already proposed in the literature. For example, TheDataTank (TDT) [27] is a tool to transform online data of different formats into a user-friendly RESTful API, where the data owner provides the URI base of her data and the meta-data needed to interpret the data.

However, IES Cities focuses on offering an integrated solution that solves not only the data source's metadata management but also provides solutions for other Smart Cities related aspects: data source access with an homogeneous query language, analytics services for publishers, and an application management infrastructure to foster the participation of developers into the Smart City's ecosystem.

During the last years, Smart Cities have integrated different solutions in order to solve some of the previous problems with different tools and technologies. For example, CKAN [18] is system mainly focused on managing metadata in a similar way to traditional Content Management Systems (CMS). It contains functionality to enable consumers to discover useful datasets, using a tag system or a keyword based search. This is a similar approach to the one followed by IES Cities, however, our solution not only provides mechanisms to find data sources but also to query them with a common language that eases consumers, mainly developers, to consume and also contribute to the city's data. CKAN also provides a simple mechanism, through an extension known as the DataStore, which enables to query data stored in tables managed by the platform. However, the query language used by CKAN's DataStore lacks the expressivity of the language used in our solution and is only applicable to data in tabular format.

Socrata [21] is a proprietary platform that provides a similar solution to CKAN but, while CKAN mainly focuses on data source's metadata management, it also solves the data storage problem, providing a more integrated solution. However, IES Cities focuses also on providing access to already existing datasets, in order to allow a gradual adoption of the technology.

Finally, Apache Stanbol [22] allows extending other CMSs with semantic technologies and integrate them with rules and reasoning. It leverages on the existence of a *FactStore* component. Data consumers and producers can query and update this store with facts, using an SQL-like language. However, as in the previous cases, it does not allow connecting with already existing data sources, as our proposed platform does. In addition, IES Cities also provides a query mechanism for SPARQL [31] data sources, oriented to ease the access to this type of datasets, which are not commonly known by application developers.

Our solution has some resemblances to the Data Hub's module of MK:Smart [25], which enables the collection, integration and usage of data from diverse sources of city systems. This module also includes a data catalogue that supports the description and traceability of the data for the generation of analytics. MK:Smart's architecture uses different smaller storage components, which contain data in different formats: RDMS, NoSQL, Triple Store or Graph DBs, supported by some wrappers. The data integration is extended with HyperCat [26], which is a solution to integrate different IoT systems inside MK:Smart.

However, our solution uses an extendedly known query language, i.e., SQL to query the data sources, in order to ease the access to all common structured data: CSV, JSON, RDF endpoints or relational databases. Furthermore, we could also extend our platform to support other data formats and sources, for example, those supported by HyperCat, by connecting it to the IES Cities solution with the usage of data translators.

In addition to the previous technologies, there are also other proposals for the development of Future Internet applications applied to Smart Cities, solving the integration of data provided by multiple sources. One example is FI-WARE [28], which is an open initiative whose goal is to create an ecosystem of ready-to use technologies, integrating sensor data and other sources. IES Cities does not try to be a competitor of the general scope provided by FI-WARE; however, the proposed infrastructure focuses on solving data access and integration issues when using open data available in different formats. Moreover, IES Cities does not have the deployment complexity required by FI-WARE as it mainly focus on the data usage required to create new apps and allow end-users and developers to collaborate in the data-gathering process.

3. The IES Cities platform

The IES Cities platform focuses on easing to developers the creation of mobile services that leverage on open data and user contributed data. In addition, one of the main goals of the project is to facilitate the deployment of data access technologies on those cities that do not have any current infrastructure. The platform achieves this goal by providing a solution that not only solves data source management for city's administrators but also focuses on reducing the gap between data consumer requirements, i.e., app developers and data providers. Furthermore, the platform fosters the reuse of already existing open datasets and data supplied by deployed sensor networks. To achieve this, the platform offers to *prosumers* (agents acting as consumers and providers) a common interface to access and contribute to the city's data.

The IES Cities platform also includes administration-oriented components, which provide city's administrators and private companies with functionality to manage the data and services usage. For example, the platform includes a module to gather statistics and analytics regarding mobile applications and datasets usage. This module is useful to the city's administrators to monitor the operation and usage of their IES Cities-compliant assets.

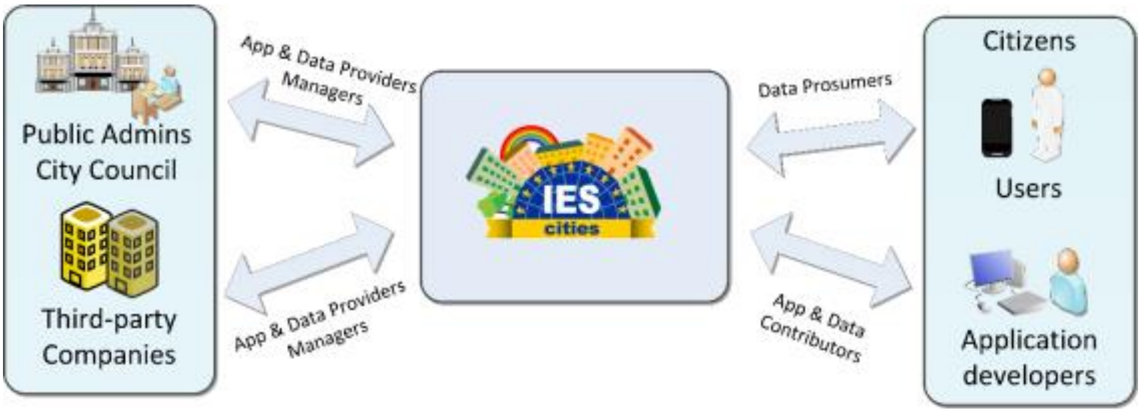


Figure 1. Agents integrated by the IES Cities vision

On the other hand, IES Cities provides administration web interfaces for all the stakeholders participating in the Smart City's data related vision: citizens, councils, data providers and developers. The roles of these agents, as depicted by Figure 1, are the following:

- ***Users/citizens*** use the IES Cities Player or the platform's web infrastructure to discover applications and data sources provided by the city or other third-party providers.
- ***City council's*** administrations, which provide open datasets and applications. They constitute the principal dataset publishers within the platform. However, third-party companies or even citizens, acting as developers or data providers, could also be interested into including their assets into the platform. Thus, city councils are also responsible for validating the quality of this published information and, therefore, the platform provides administrators' mechanisms to manage that information.

- ***Application developers*** create applications exploiting the data sources made available by city administrators or third-party providers and publish them into IES Cities and its associated marketplaces (e.g., Google Play).
- ***Third-party providers***, which can be private companies or particular citizens using the platform to publish datasets and applications. This way, not only IES Cities' vision covers public administrations, but the platform also focuses in other participants that can be active agents of the Smart City, contributing and benefiting from the possibilities provided by this new vision of the cities.

3.1. Platform architecture

This section describes the IES Cities platform's client/server architecture, which is schematized in Figure 2.

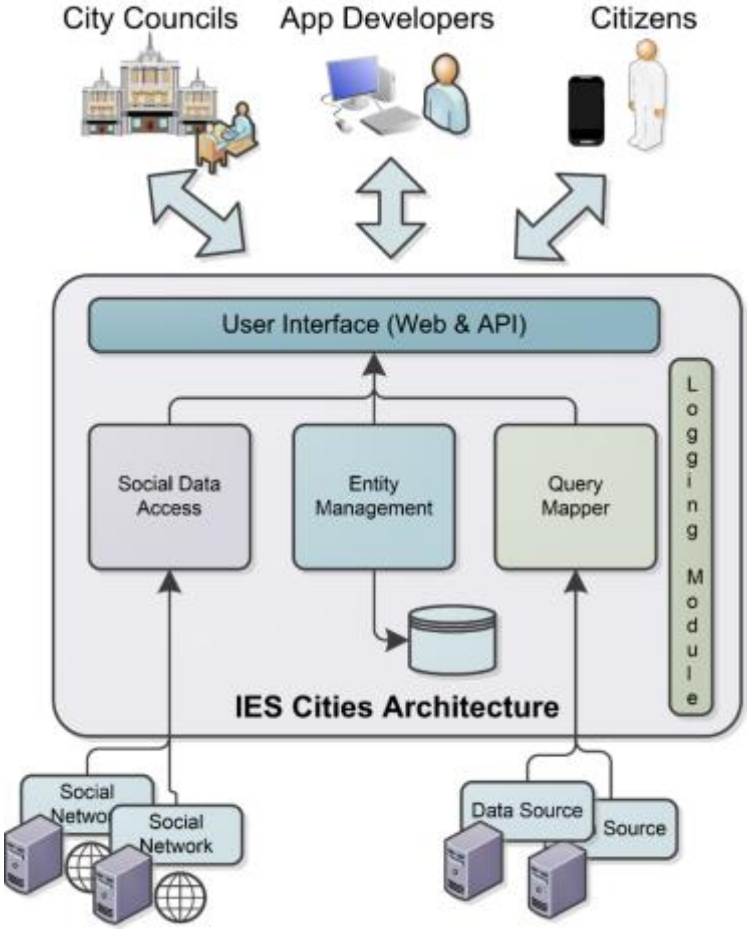
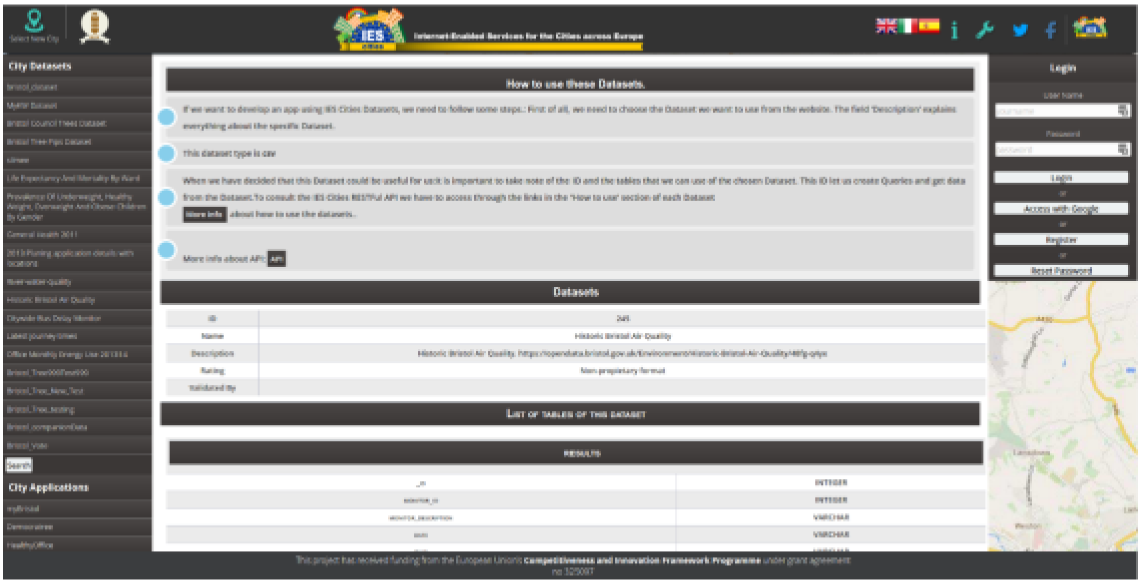


Figure 2. IES Cities' platform architecture

The architecture comprises the following components:

- **User *interface.*** The web interface allows performing different actions depending on the role of the participant agents: citizens, councils, data providers and developers. Figure 3 shows a capture of the web interface that is available to end-users and citizens to access the information provided by the platform.

- *End-users/citizens* may search and discover apps by their location inside the city or using preferences expressed as keywords, enabling them to select those apps that are more relevant to their needs in their city. Each app runs upon user selection, by either downloading or launching it (i.e., native applications). The platform also enables end-users to suggest new types of apps and datasets to which they could be interested in having access. This way, the platform creates a communication channel between the citizen and the city's public agents.
- *Developers* are provided with a list of public datasets, each dataset has an associated data model and all the configuration details (e.g., connection details and metadata about the entities modelled) that the developer needs to use. The infrastructure enables to search for datasets by description, type or format. Furthermore, IES Cities also enables developers to register, modify and review the usage statistics associated to created apps and the consumed datasets, which can be useful to create applications much closer to the citizens' needs.
- *Council administrators* use the platform to publish their applications and datasets, and to gather usage statistics about them. In addition, the platform also enables council administrators to validate and ensure the quality of applications and datasets created and published by third-party participants, enabling those external participants to be part of the public administration Smart City's vision.



How to use these Datasets.

If we want to develop an app using IES Cities Datasets, we need to follow some steps.: First of all, we need to choose the Dataset we want to use from the website. The field 'Description' explains everything about the specific Dataset.

This dataset type is csv

When we have decided that this Dataset could be useful for us,it is important to take note of the ID and the tables that we can use of the chosen Dataset. This ID let us create Queries and get data from the Dataset. To consult the IES Cities RESTful API we have to access through the links in the 'How to use' section of each Dataset

More info about how to use the datasets.

More info about API: API

Datasets

ID	245
Name	Historic Bristol Air Quality
Description	Historic Bristol Air Quality. https://opendata.bristol.gov.uk/Environment/Historic-Bristol-Air-Quality/[illegible]
Rating	Non proprietary format
Validated By	

List of tables of this dataset

RESULTS

_id	INTEGER
monitor_id	INTEGER
monitor_description	VARCHAR
date	VARCHAR

This project has received funding from the European Union's **Competitiveness and Innovation Framework Programme** under grant agreement no 325097

Figure 3. IES Cities web interface

- **Entity management.** This component provides access and management of the different entities provided by the platform: users, datasets, councils and applications, not only through an end-user web interface but also through a programmatic interface (i.e., REST API) enabling external developers to contribute. Section 3.2 provides a more detailed description about these entities.
- **Logging module.** It offers functionality through which applications can register both platform and app specific events. Developer can later exploit this information to extract conclusions about the usage of the platform and its data sources, which can be an extremely useful resource for council administrators when developing their city's data and services strategies.

- **Social data access.** This module enables the platform to access social networks such as Facebook or Twitter, which, in a more interconnected Smart City, can provide new sources of information that enhances the data directly provided by administrations or third-party agents.
- **Query mapper.** The component provides a common interface to access heterogeneous data sources. It maps queries and updates to each dataset's underlying data format with the help of data mediators.
- **IES Cities player.** This component, which is an application running on the citizens' smartphones, is responsible for the information presentation and user interaction, allowing service searching, browsing and execution. It also provides mechanisms to provide feedback about apps experience and demands for new functionality to the platform managers.

3.2. IES Cities managed entities

Internally, the IES Cities platform manages data about the following entities:

- ***Application.*** These entities represent IES Cities applications registered either by council administrations or by third-party developers. Their main properties are the name of the service, a description about them, and its download location (i.e., a pointer to the App Marketplace where they may have been registered), which enables to syndicate IES Cities with the Google Play market. Each service can also be associated with a certain geographical scope to allow citizens to discovery applications based on the user's detected location, in addition to a basic keyword search.
- ***Dataset.*** These entities contain information about each data source registered within the platform. Each dataset entity contains the following details: id, description, type (CSV, JSON, SPARQL, and DB) and other information about is structure and permission data, which the Query Mapper uses to connect with the data source.
- ***Council.*** These entities represent the managed councils registered within the same IES Cities instance. The inclusion of the councils as a first-class entity into the platform allows IES Cities to be deployed and managed by the own city's council or as a solution that provides functionality for different councils. In the latter case, this allows that a region administration or a third-party agent can maintain the platform.
- **User.** These entities maintain information about the different users and roles that can play in the platform: citizens, council admins and application developers/publishers. In order to allow a broader usage of the platform, it is optional for citizens to register themselves in the platform to discover applications or consume the information offered by the datasets. However, registration is required to publish data or change entities related information.

4. Query Mapper module

The Query Mapper module of the IES Cities architecture provides the functionality to perform queries and updates on registered datasets and transform the results to a common response format. The main objective of this module is to provide developers with an easy to use data access mechanism, which hides the differences and complexities of the specific format of the registered data sources.

In the growing ecosystem of open data sources in Smart Cities, it is common to find heterogeneous sets of data formats with different access/update methods that application developers are required to learn to fully exploit the potentials of available data. For example, in the case of a SPARQL data source, developers are required to understand the subtleties of semantic technologies (OWL, RDF) and the particular characteristics of its query language. On the other hand, the usage of JSON or CSV data sources requires users and applications to download the full dataset and perform the data selection process locally by the user's app.

Thanks to this component, IES Cities aims to reduce the gap between developers and the data they want to use in their applications, not only when consuming data but also when creating applications that allow users, or connected sensors, to contribute with new information to the Smart Cities ecosystem. In order to provide a common vision of the different data sources available in the city, the module performs a mapping of the current data source's structure to a relational view. The module allows consumers and producers to access data using SQL, a more usually known query and update language by developers.

The Query Mapper, through the IES Cities infrastructure, provides the following high-level functionalities to the Smart City's data consumers and producers:

- **Data query.** This functionality enables querying a compatible data source. The query mapper provides methods to query data using the SQL query language. The queried data source must be compatible with these types of queries, which means that the dataset is already relational (database) or that the Query Mapper maps it into a compatible relational schema. Supported data source types, transformable to a relational view, are *SPARQL*, *JSON* and *CSV*.
- **Semantic data query.** This feature enables querying a SPARQL compatible dataset, which means that more experienced developers can use it to access resources in RDF format. Semantic query can be useful in those situations that require exploiting the benefits of semantics technologies and Linked Data. In this case, only data sources backed by SPARQL endpoints are available through these query language. The Query Mapper acts only as a translator with the SPARQL endpoint, which can be useful to provide a common access mechanism for all the data sources in the Smart City through the IES Cities platform and to reduce the complexity that SPARQL endpoints impose to application developers.
- **Data update.** This method provides the functionality to insert, update or remove data from the registered data sources using SQL update clauses. Data sources can be available as JSON or CSV, usually static read-only files. In other cases, third-party agents provide some data sources that do not allow the direct update of the data source. The Query Mapper solves these situations by creating local data stores in the IES Cities platform instance to store user provided data. In addition, the update methods also provides functionality to update datasets using a JSON object describing the changes to the data. This can be useful for those developers that want to use JSON as their communication format not only for information retrieval, but also to perform data updates.

4.1. Internal architecture

There are two different use case scenarios covered by the Query Mapper, consumers and producers of information, although it is also possible that the same client could fit both roles acting as a prosumer of information. Figure 4 depicts the internal architecture of the Query Mapper module.

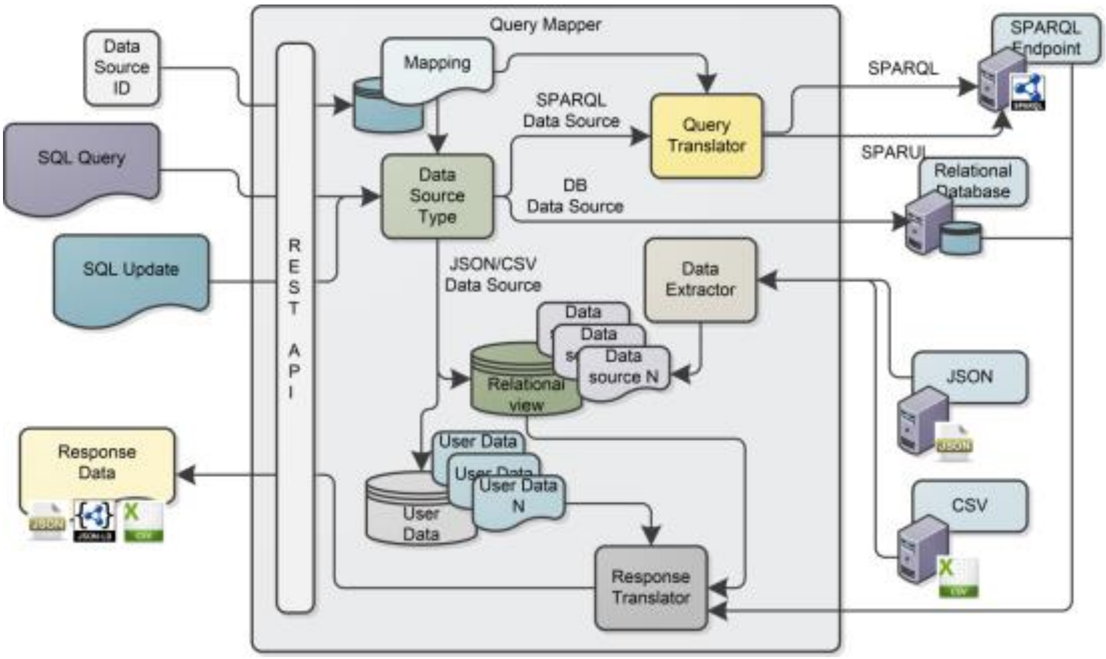


Figure 4. Query Mapper internal architecture

If the client acts as a data consumer the process begins with the client sending a SQL query to the Query Mapper. Then, the module sends the query together with the registered data source's identifier, which the Query Mapper uses internally to retrieve the associated mapping. In order to improve the clarity of the diagram. Figure 4 partially represents the data source registration, mapping storage and retrieval processes.

After obtaining the *mapping* description and, depending on the data source type, the Query Mapper performs the required transformations and/or redirects the query to obtain the data from the corresponding source. Currently, the IES Cities module supports four data source types: SPARQL, JSON, CSV and relational databases.

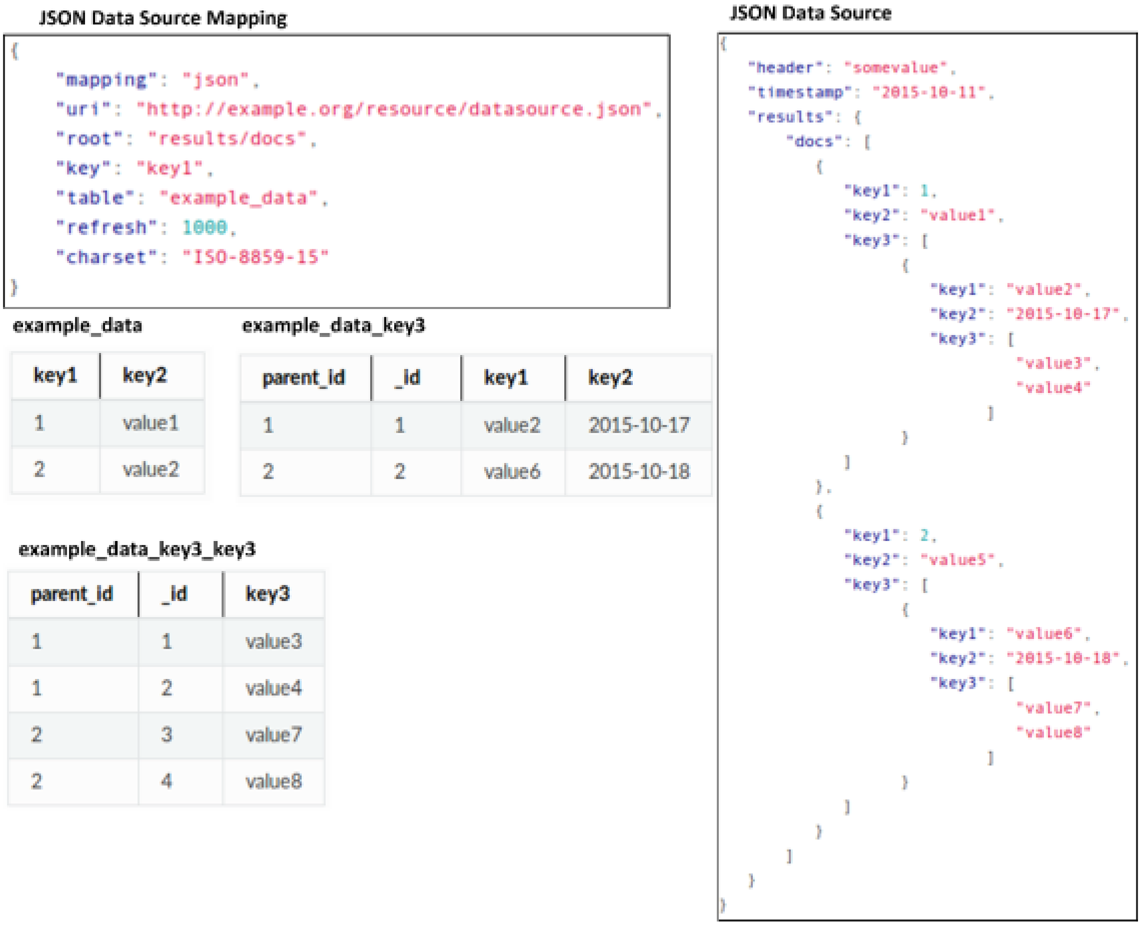
- **SPARQL.** The unified mechanism allows accessing to data sources available at SPARQL endpoints to query and update them. In this case, the module transforms SQL queries and updates to their SPARQL or SPARUL (SPARQL Update Language) equivalent statements. Then, it executes these translated statements against the associated endpoint.
- **JSON.** The Query Mapper can access any resource (web or local) available as JSON file. The JSON data source is analysed and its schema extracted to provide a relational view of the data. Section 4.2 provides more information about the schema and data extraction processes. As this type of data source is usually available as a static file or provided by a database management system that does not allows updates, the Query Mapper provides support to store the user updates locally within the IES Cities platform. The module adds this user data with the same schema as of the original JSON file. In addition, to control the provenance of the data, the Query Mapper annotates users' contributed data to allow future consumers to distinguish it from the original source's one.
- **CSV.** Any resource (web or local) accessible as a CSV file is also connectable within the Query Mapper. The process is similar to the one explained before for JSON data sources, however, in this case the relational view is obtained from the CSV structure.

- **Database.** The module also offers the possibility to connect with relational databases managed by a RDBMS (particularly PostgreSQL, MySQL or SQLite). With relational databases, which are directly compatible with SQL queries, the Query Mapper only redirects the query, or the update, to the corresponding database manager. The IES Cities platform provides this functionality to offer a mechanism to facilitate the connection of legacy data sources within the IES Cities platform, easing the migration to the Smart City's open data ecosystem.

Finally, as show in Figure 4, the Query Mapper manages two different internal storages: the first one contains relational representations of the schema and data of JSON and CSV datasets; while the second one stores user data, added or modified by updates from other users, which can be queried as any other data source.

4.2. JSON data sources

The Query Mapper requires some information to access data sources in JSON format. Figure 5 shows an example of the *mapping* description that is required by the module to connect with a JSON file. The current implementation of the Query Mapper uses JSON as the language to express mappings according to the canonical form adopted within IES Cities.



JSON Data Source Mapping

```
{
    "mapping": "json",
    "uri": "http://example.org/resource/datasource.json",
    "root": "results/docs",
    "key": "key1",
    "table": "example_data",
    "refresh": 1000,
    "charset": "ISO-8859-15"
}
```

example_data

key1	key2
1	value1
2	value2

example_data_key3

parent_id	_id	key1	key2
1	1	value2	2015-10-17
2	2	value6	2015-10-18

example_data_key3_key3

parent_id	_id	key3
1	1	value3
1	2	value4
2	3	value7
2	4	value8

JSON Data Source

```
{
   "header": "somevalue",
   "timestamp": "2015-10-11",
   "results": {
       "docs": [
          {
             "key1": 1,
             "key2": "value1",
             "key3": [
                {
                   "key1": "value2",
                   "key2": "2015-10-17",
                   "key3": [
                      "value3",
                      "value4"
                   ]
                }
             ]
          },
          {
             "key1": 2,
             "key2": "value5",
             "key3": [
                {
                   "key1": "value6",
                   "key2": "2015-10-18",
                   "key3": [
                      "value7",
                      "value8"
                   ]
                }
             ]
          }
       ]
   }
}
```

Figure 5. JSON data source example mapping and resulting relational tables

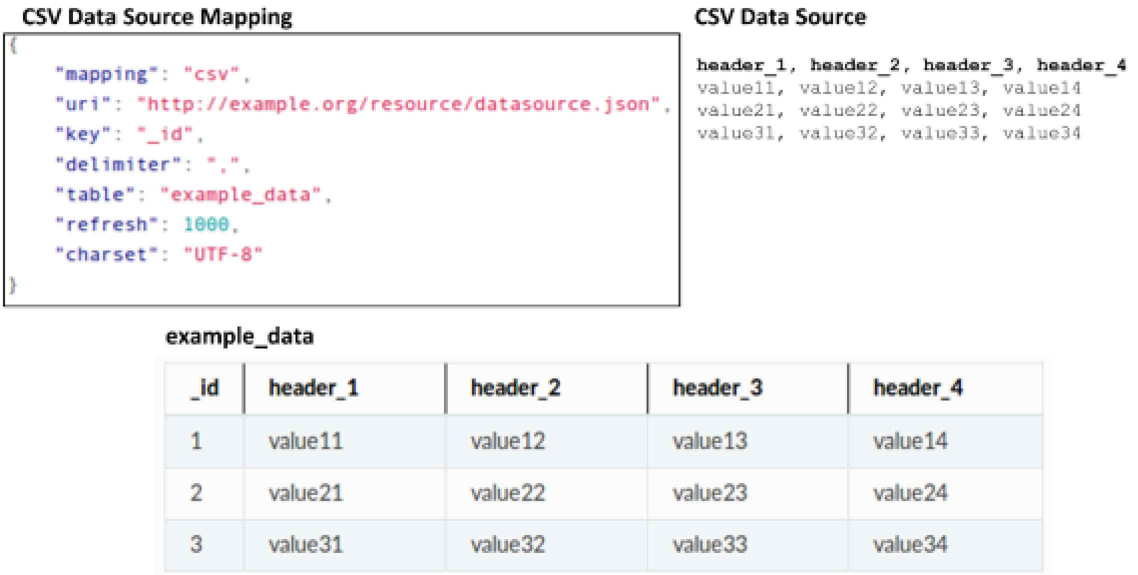
Using the information provided by the mapping description, the Query Mapper connects to the resource containing the data, in this case a JSON file, downloads the data and analyses it, extracting its schema in order to create a relational view of the data source, which is populated with a copy of the data. The Query Mapper extracts the JSON structure and creates tables representing the same data structure. As an example of the relational view creation, Figure 5 also shows the tables generated as result of the schema extraction and data population.

In the current Query Mapper's implementation, the extracted tables are stored inside an RDBMS (i.e., SQLite) internally managed by the module, which will execute the SQL queries send by the users and obtain the required data in the selected format.

4.3. CSV data sources

When the Query Mapper connects to a CSV data source, the process is much simpler than the one explained in Section 4.2; the structure of CSV files does not allow the introduction of nested objects, with a known common structure as JSON does, and they directly resemble the structure of a simple table with multiple columns. Although these columns can contain complex data (e.g., a list of space-separated values); their structure is data dependant and it is not easy to extract the data into multiple tables, as in the case of JSON data sources.

Figure 6 shows an example of the mapping description required to connect the Query Mapper with data sources in CSV format. In this case, the mapping attribute has the "csv" value, which the module uses to identify the data source as a CSV resource.



CSV Data Source Mapping

```
{
    "mapping": "csv",
    "uri": "http://example.org/resource/datasource.json",
    "key": "_id",
    "delimiter": ",",
    "table": "example_data",
    "refresh": 1000,
    "charset": "UTF-8"
}
```

CSV Data Source

```
header_1, header_2, header_3, header_4
value11, value12, value13, value14
value21, value22, value23, value24
value31, value32, value33, value34
```

example_data

_id	header_1	header_2	header_3	header_4
1	value11	value12	value13	value14
2	value21	value22	value23	value24
3	value31	value32	value33	value34

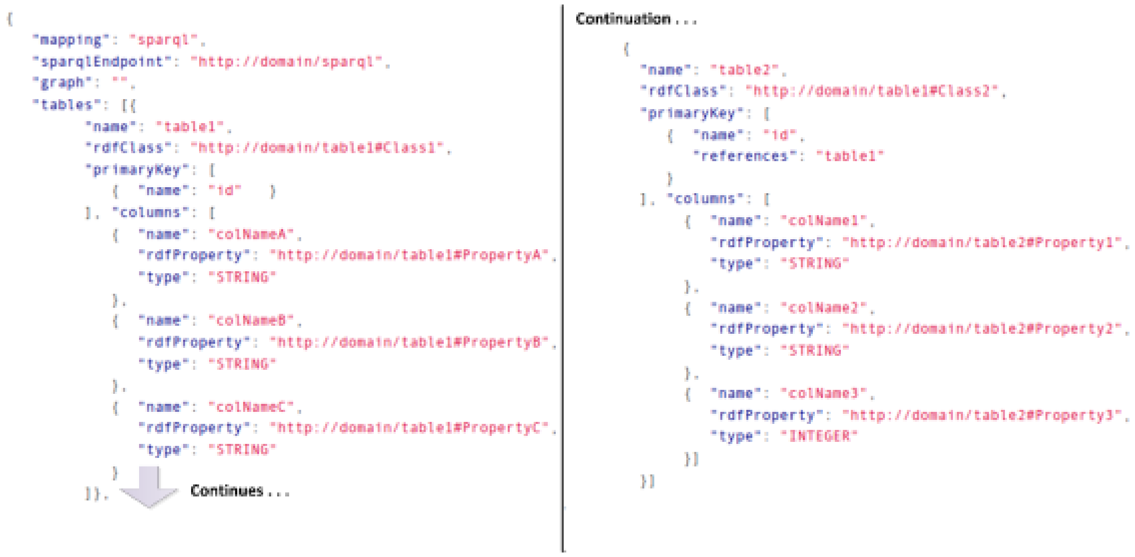
Figure 6. CSV data source example mapping and resulting relational tables

The example shown in Figure 6 represents a situation where a simple CSV file provides the data. The Query Mapper extracts the schema and the data, creating a new table with the name specified by the mapping description containing the data provided by the CSV data source. The module obtains the type of the table columns by analysing the type of the values contained in the CSV for each of the columns, following an identical procedure as the one applied for JSON resources. The resulting table containing the extracted data is stored into a local RDBMS, which allows the direct execution of SQL queries against the transformed data.

4.4. SPARQL data sources

The Query Mapper also allows connecting with SPARQL endpoints, which is useful to provide a simpler mechanism to access data that is accessible only in RDF format. In this case, the required mapping is more complex than the one used for JSON and CSV data sources, explained in the previous section, because the Query Mapper needs some extra information to map between the relational and semantic view of the data.

In this case, the mapping description for a SPARQL endpoint expresses the connection between the relational view and the RDF model. It contains the correspondence among the classes and properties of the semantic view and the tables and attributes of the relational one. Figure 7 shows an example of the mapping file required to connect with SPARQL endpoints.



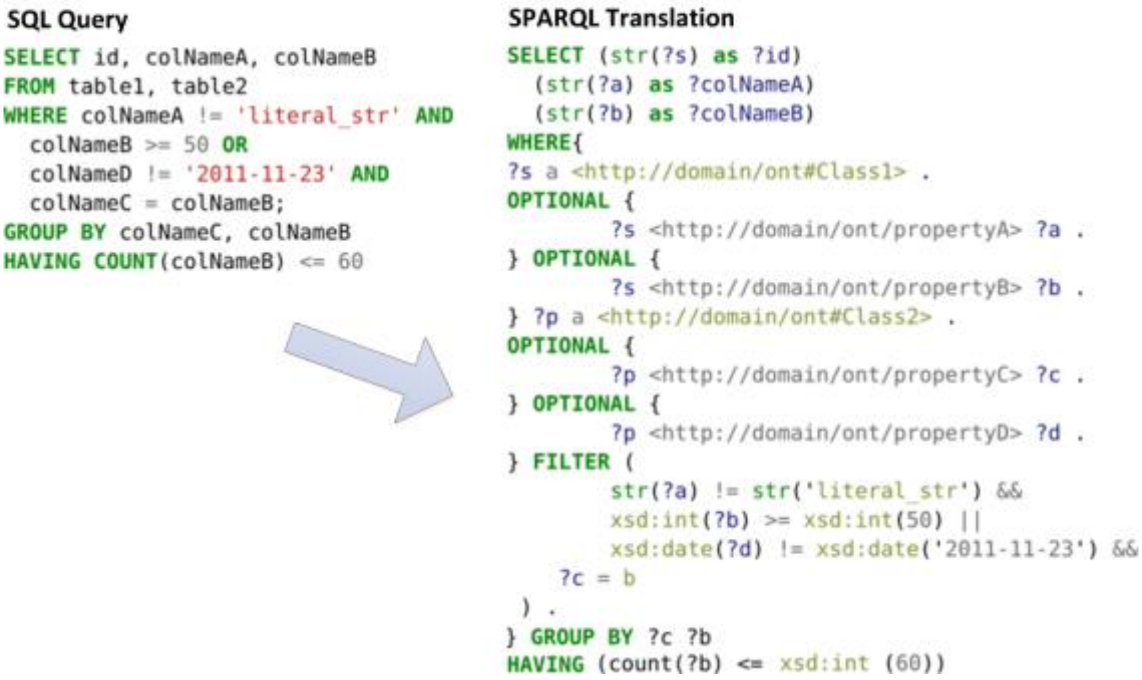
```
{
   "mapping": "sparql",
   "sparqlEndpoint": "http://domain/sparql",
   "graph": "",
   "tables": [{
         "name": "table1",
         "rdfClass": "http://domain/table1#Class1",
         "primaryKey": [
            {  "name": "id"   }
         ], "columns": [
            {  "name": "colNameA",
               "rdfProperty": "http://domain/table1#PropertyA",
               "type": "STRING"
            },
            {  "name": "colNameB",
               "rdfProperty": "http://domain/table1#PropertyB",
               "type": "STRING"
            },
            {  "name": "colNameC",
               "rdfProperty": "http://domain/table1#PropertyC",
               "type": "STRING"
            }
         }],
```

Continues . . .

Continuation . . .

```
      {
        "name": "table2",
        "rdfClass": "http://domain/table1#Class2",
        "primaryKey": [
           {  "name": "id",
              "references": "table1"
           }
        ], "columns": [
             {  "name": "colName1",
                "rdfProperty": "http://domain/table2#Property1",
                "type": "STRING"
             },
             {  "name": "colName2",
                "rdfProperty": "http://domain/table2#Property2",
                "type": "STRING"
             },
             {  "name": "colName3",
                "rdfProperty": "http://domain/table2#Property3",
                "type": "INTEGER"
             }]
        }]
```

Figure 7. SPARQL endpoint mapping description



SQL Query

```
SELECT id, colNameA, colNameB
FROM table1, table2
WHERE colNameA != 'literal_str' AND
  colNameB >= 50 OR
  colNameD != '2011-11-23' AND
  colNameC = colNameB;
GROUP BY colNameC, colNameB
HAVING COUNT(colNameB) <= 60
```

SPARQL Translation

```
SELECT (str(?s) as ?id)
  (str(?a) as ?colNameA)
  (str(?b) as ?colNameB)
WHERE{
?s a <http://domain/ont#Class1> .
OPTIONAL {
        ?s <http://domain/ont/propertyA> ?a .
} OPTIONAL {
        ?s <http://domain/ont/propertyB> ?b .
} ?p a <http://domain/ont#Class2> .
OPTIONAL {
        ?p <http://domain/ont/propertyC> ?c .
} OPTIONAL {
        ?p <http://domain/ont/propertyD> ?d .
} FILTER (
        str(?a) != str('literal_str') &&
        xsd:int(?b) >= xsd:int(50) ||
        xsd:date(?d) != xsd:date('2011-11-23') &&
    ?c = b
  ) .
} GROUP BY ?c ?b
HAVING (count(?b) <= xsd:int (60))
```

Figure 8. SQL to SPARQL translation example

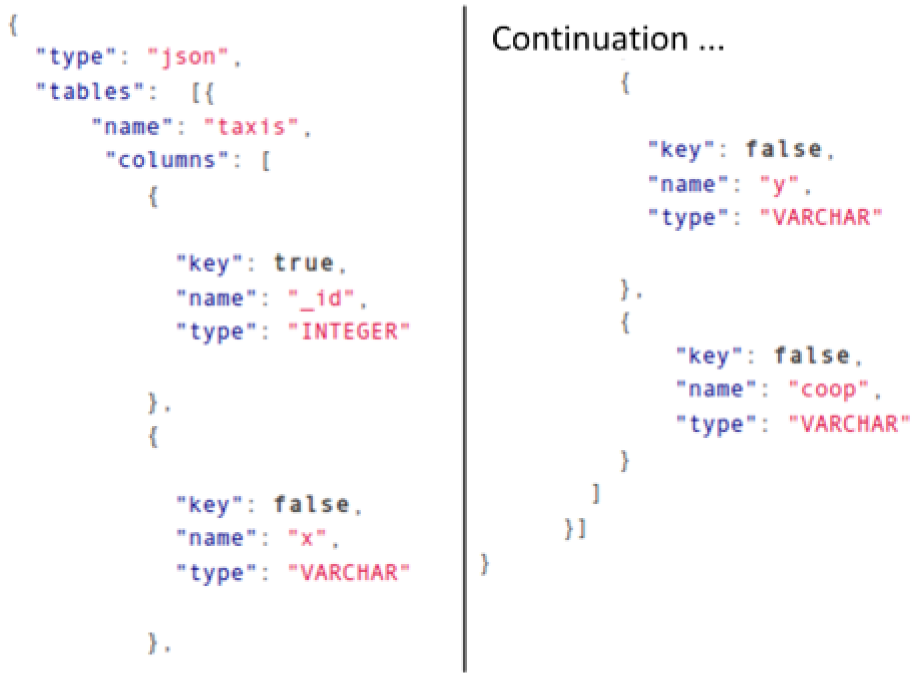
The Query Mapper module transforms, as shown by Figure 8, the SQL queries to an equivalent SPARQL query by parsing the string submitted by the user. The resulting sentence is directly executable on the selected endpoint. The parser is implemented with ANTLR [30], using the information provided by the mapping description. Currently, the translation process supports SELECT statements and the following clauses: ALL, DISTINCT, WHERE, GROUP BY, HAVING, ORDER BY and LIMIT, which, according to the performed evaluations, are enough to fulfil the client needs when retrieving data.

5. Data access management

This section covers the process that users follow in order to use the data provided through the IES Cities infrastructure. The data source publication requires the definition of a mapping description, which, usually, the data publishers or the administrators of the IES Cities platform will create. When the publisher registers the mapping into the platform, the Query Mapper module analyses the structure of the data source and constructs the relational view of the connected data sources.

5.1. Discovering data structure

The processes of data source discovery and selection start by using the web interface or the programmatic API that the IES Cities platform offers. Users trying to consume the data need to know the structure in order to adequately query the data source and retrieve the desired information. Once a consumer identifies a useful data source, the infrastructure provides functionality to obtain the structure of the data source connected to the platform.



```
{
  "type": "json",
  "tables":  [{
      "name": "taxis",
       "columns": [
         {

           "key": true,
           "name": "_id",
           "type": "INTEGER"

         },
         {

           "key": false,
           "name": "x",
           "type": "VARCHAR"

         },
```

Continuation ...

```
         {

           "key": false,
           "name": "y",
           "type": "VARCHAR"

         },
         {
             "key": false,
             "name": "coop",
             "type": "VARCHAR"
         }
       ]
    }]
}
```

Figure 9. Data source description provided by the platform

As the IES Cities platform focuses on providing a relational view of all connected data sources, i.e., JSON, CSV, SPARQL and DB sources, users can homogenously explore the structure of these sources for all supported types. The IES Cities platform provides simple API methods to obtain a description of the data source’s tables, columns and data types. This functionality is used by the IES Cities solution's web interface, shown in Figure 3, but can also be used by other third party applications to search for compatible data through the available sources. **Error! Reference source not found.** shows an example of the information used to describe a registered data source.

5.2. Data response formats

As one of the main goals of the IES Cities platform is to facilitate the creation of urban apps, it is useful to provide different formats for the responses obtained when querying any of the sources connected through the Query Mapper module.

Although not all formats are compatible with every data source, users can select the response data obtained after querying the data sources in any of the following formats:

- **JSON.** This is the default response format used by the Query Mapper module when consulting any of the data source types supported by the Query Mapper module, i.e., JSON, CSV, SPARQL or DB. Nowadays, JSON is considered a *de-facto* standard when transporting data across the Internet, mainly for its simplicity for developers and its consumption in mobile and web applications.
- **JSON-LD.** This format is a JSON serialization of Linked Data that also provides semantic metadata information. Its main feature is that it combines the simplicity of JSON with the power of Linked Data, allowing consumers to link the data with other sources and increasing the utility and applications of the Smart City's data. In this case, the Query Mapper can only obtain this data response format from SPARQL data sources, which are the ones that contain the additional required metadata.
- **CSV.** Due to the table structure of response data, they are transformable to a simple CSV response. The Query Mapper provides this format to improve compatibility among existing data sources and to allow users to easily download the data and import it directly into data management or visualization tools.

5.3. User provided data

The IES Cities project aims at enabling citizens to participate into the process of Smart City's data consumption but also into gathering and producing their own data. For this reason, the IES Cities platform provides, through the Query Mapper, the mechanisms to allow clients the possibility to contribute with their data to available data sources.

As shown in Figure 4, the Query Mapper module integrates an internal data storage aimed to contain the information provided by users. The module intentionally separates the user data from any other information extracted from the original sources, allowing the management of the data provenance for quality assurance or legal purposes.

Therefore, this separation of the citizen provided data from that contained in the original data source aims to solve two important problems in the ecosystem of consumers and providers of the Smart City:

- **Provenance.** Consumers must clearly know the origin of the data; they should be able to distinguish if the data is provided and validated by official sources or, on the other hand, if the data is the result of some other citizen contributions. The IES Cities platform achieves this by storing citizen contributions in a separate storage and annotating the responses containing user provided data. The platform can provide the provenance information to consumers using a standard format, such as the WRC PROV model [29].

- **External data integration.** Some already available data sources could exist in the Smart City's open data ecosystem. IES Cities focuses on integrating this data into the city's open data platform, and achieves this vision thanks to the mapping functionality offered by the Query Mapper. In some cases, it could not be possible for users to update the original data source, due to permission restrictions or other limitations as, for example, those data sources offered as static JSON or CSV files with no update mechanisms. In those cases, the IES Cities platform stores user's contributions, enhancing data sources that could not allow participation in any other way.

To achieve a full control of data provenance, the module features a permission system that allows controlling access rights to citizen-provided data. This permission system enables a fine control of the insert, delete and update processes of the citizen/user data stored into the platform. The Query Mapper module integrates the data access control into the citizen data storage by adding metadata to the information inserted by the users.

The basic functionality of the permission system requires users to register in order to contribute with their data. Urban application developers can facilitate this process, using the user management mechanism provided by the IES Cites platform. In addition, different data control policies can be applied; for example, user inserted data could be accessible to any user, registered or not, or only to some subset of registered users/groups. In addition, data modifications can be restricted in the same way to some subset of the registered users.

5.4. Data origin selection

As the user provided data is stored separately from the data source's original one, the Query Mapper allows users to select the origin of the data. The following alternatives exist:

- **Original:** the query mapper executes the query on the external store containing the data obtained from the connected data source.
- **User:** the module executes the query on the user data store, which contains data contributed by IES Cities' users.
- **All:** in this case, the module obtains the data from the two different storages: original and users data. The module merges the results of those two queries and return them to the users as a single response. In addition, the module marks user provided data in the response to show its provenance. As the data update mechanism is only available to registered users, response data contains for each queried row the identifier of the user that inserted that data into the data source.

6. Validation

The IES Cities platform is part of the work developed by a European project, during the duration of the project it has been deployed on a local network to evaluate its usability, and technical aspects, for users (service consumers), developers (service providers) and city councils (data providers).

IES Cities introduces new processes for the management of data and its related services, not only for developers but also for city councils. Therefore, we have validated the platform taking into account the different roles, identified in Section 3, which stakeholders take to participate.

6.1. City council participation

We have validated the participation of city councils as primary data providers thanks to the participation of four different cities with different degrees of maturity in their open data strategy. For example, while Zaragoza and Bristol already had previous experience on providing data and urban applications to their citizens, Majadahonda and Rovereto have used the experience and results of the project to start and take impulse on their data strategy.

Therefore, the project has tackled different situations when trying to integrate the different situations of each city, which are summarised in the following way:

- **Zaragoza.** As this city already provided an strong data platform to their citizens, the IES Cities platform has been primary tested as a new application development infrastructure, which provides new utilities to developers to easy the access to already existent data sources. Section 6.2 provides a summary of the results obtained when validating the platform with application developers, which use the capabilities of the Query Mapper to access the data.
- **Bristol.** The deployment of the IES Cities platform within this city has followed a similar approach to the previous one. As this city already provided multiple data sources to their citizens, already used to develop urban applications, the IES Cities infrastructure was validated by connecting existing data sources to the platform.
- **Rovereto.** The IES Cities infrastructure has enabled the city to start working on their open data strategy by creating and connecting new data sources, exploitable through the platform.
- **Majadahonda.** In this case, the city did not have any defined open data and urban service strategy. Therefore, the IES Cities platform has served as the primary platform to publish the data sources and made them available to developers.

6.2. Third-party developer participation

IES Cities mainly focus on improving the development of urban applications by city councils but also fostering the participation of external developers: third-party companies or citizens that want to contribute to their city's infrastructure.

In order to evaluate the participation of third-party agents, IES Cities' project partners have organised different events oriented to implicate developers into the creation of urban applications. Specifically, some of these events have focused on the organization of hackathons oriented to the development of urban application using the Smart City's data and the infrastructure provided by IES Cities. During and after these events, developers were interviewed (up to 90 developers in the city of Zaragoza) to collect their opinions about the new data access paradigm provided by the IES Cities platform and comparing it to some previous mechanism that provide a heterogeneous access to the data.

The results of this developer evaluation process can be summarized as follows:

- **API complexity.** The majority of the interviewed developers think that the REST approach followed by the platform is easy to apply and use in their applications, despite some bugs and problems found during its public release as it was the first time that the platform was open to citizens.
- **Homogenous data access.** Developers also provided good critics about the homogenous data access to different data source formats using SQL. Almost every development team during the hackathon had knowledge about the basic usage of the SQL query language and, therefore, they used it broadly to access data sources in different formats: CSV, JSON and SPARQL endpoints. In the latter case, specific

interviews with developers showed that, although SPARQL is a well-known language inside the academia when using semantic information, it does not have the same acceptation between general application developers. Therefore, simplifying the access to data sources by providing a more traditional data access mechanism, such as SQL, is a good approach.

- **User data publication.** During the events, developers did not only create applications having a passive role in their city's data consumption processes, but also took part by developing applications which allow final users to contribute to the datasets. IES Cities allows developers to create applications that let users to participate as *prosumers* using simple SQL update sentences.

During the hackathons, developers also raised attention to some issues that, at that moment, the IES Cities platform was not covering:

- **User personal data source creation.** The initial approach of IES Cities was that councils or third-party related agents could create and publish the data sources. Later, applications and citizens consume these data sources, allowing the contribution to their data. However, a common issue was that some developers required not only using or contributing to existing data sources but also to create some new data sources from scratch, which their newly designed urban apps will fed with information. This functionality was, therefore, included into the Query Mapper. This way, developers can create specific datasets for their applications by providing a sample JSON description of the data format. Then, the module applies a process identical to the one explained in Section 4.2, generating a local database that allows storing data. These personal data sources can configured as public or private, depending on their associated permissions. The IES Cities platform marks personal data sources as user provided, which enables to differentiate them from the original information provided by city councils, as explained by Section 5.
- **Data federation.** Currently, the IES Cities platform does not provide a mechanism to perform distributed queries and federate data coming from different data sources. In its current implementation, the clients must solve this problem by adding more complexity to their applications' logic. However, as all the data managed by the IES Cities platform provides a common relational approach, the platform can solve the federation of multiple sources in a similar manner to that the one followed by distributed databases.

6.3. Application development proof-of-concept

In addition to the validation by third party developers explained in Section 6.2, the creation and development of 16 specific applications, by each of the participant cities and other associated developers, has guided the design of the IES Cities platform by demonstrating the possibilities of the data access infrastructure.

Particularly, this section summarises two applications used to validate the data access module of IES Cities. The first one, Zaragoza Complaints & Suggestions, is a mobile service developed as one of the urban applications that are part of the IES Cities project. On the other hand, Bicicas is an application developed externally to the project with the aim to validate the usability and utility of the proposed infrastructure.

Zaragoza's Complaints & Suggestions. This first service, shown in Figure 10, uses open data to get an overview of the reports and faults occurring in the city's public infrastructure. The application demonstrates how a developer, with the help of the IES Cities platform, can create mobile web services relying on semantic data without technical knowledge of the query and data modelling language in this case, SPARQL and RDF, respectively. Thanks to the support of the IES Cities platform, a web developer only needs to create a query in the standard SQL language and send it to the Query Mapper.

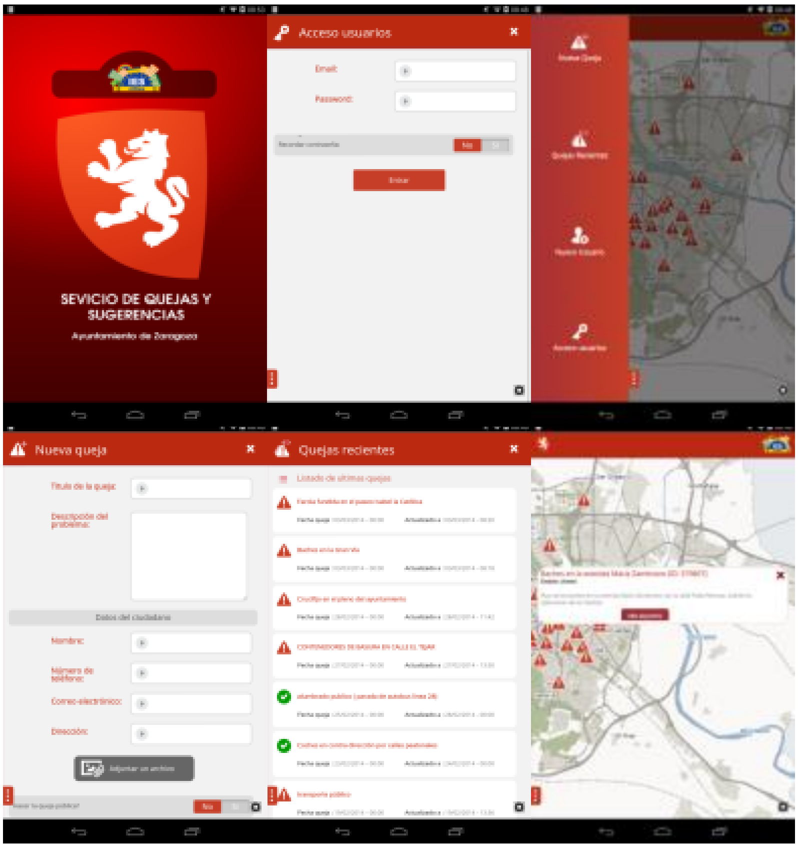


Figure 10. Zaragoza Complains & Suggestions.

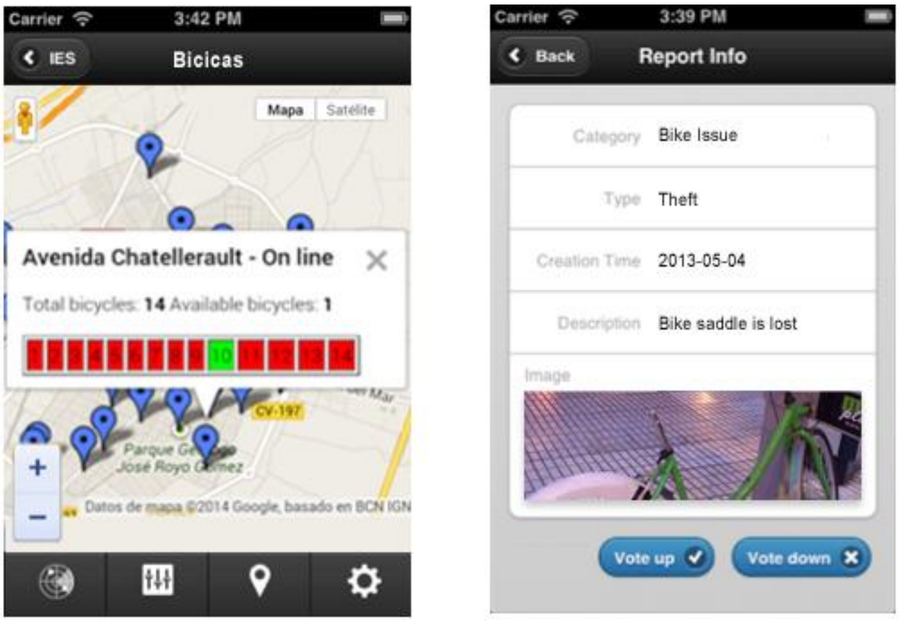
From left to right Figure 10 shows the splash screen, login page, map view of complaints and suggestions with menu, new complaint and suggestion form including title, description and optional user details, recent complaints listing.

Actually, this service defines both a data entry point and a service entry point. The first one enables the application to submit SPARQL queries the connected endpoint, which can query the encoded data in GeoReport[3] format, XML derivative containing information about city's infrastructure. Since Zaragoza's council wants to comply with standards, such as Open311 [15], but it also wants to commit towards open data of the highest standards, the system maps GeoReport records into RDF triples representing the same info in a semantic form. The translator modules explained in Section 4 achieves this transformation.

The fact that developers need only to submit SQL queries through a REST API to the IES Cities Query Mapper demonstrates the IES Cities platform capabilities to accelerate urban app development. The IES Cities platform's component talks to the Zaragoza SPARQL endpoint and maps the results in JSON so that the Zaragoza Complaints & Suggestions app can more easily processed them.

[3] http://wiki.open311.org/GeoReport_v2

Castellón's Bicicas. The second service, Bicicas[4], demonstrates how the IES Cities platform can integrate already existing urban services by using the functionalities provided by the Query Mapper. The original Bicicas service provides data as HTML web pages consumable by human beings. The information contains the state of all different bicycle-sharing points within the city of Castellón (Valencia, Spain); each sharing point contains an array of docks for locking each bicycle, the state for each dock can be bicycle available or bicycle unavailable.



Avenida Chatellerault - On line

Total bicycles: 14 Available bicycles: 1

Category: Bike Issue

Type: Theft

Creation Time: 2013-05-04

Description: Bike saddle is lost

Figure 11. Bicicas sharing bikes service.

Figure 11 shows on the left a map available to check the state of each sharing point, while on the right it shows the mechanism used to report issues about the state of a bicycle.

The application was developed using this sharing point information to create a set of new RESTful services that allow checking the availability for all sharing points. In addition, to provide the location of each sharing point in the city, the service uses microdata, i.e., *georss*[5]. This microdata also allows enriching the original data, meaning that a wider range of applications can use it instead of being only available for human consumption. For instance, Google Maps can directly render Atom sources as a map layer.

Users can check bicycles availability at any sharing point in the city and report any issue about the sharing services and the state of the bicycles. Figure 11 shows the main window of the mobile application. By interacting with the map, a user can check the number of bikes available at each sharing point and submit an issue about any state of the bicycles. These services are reusable by means of the TDT or CKAN to deploy new services by selecting a few items from the original services, or to add some semantics and to link them with sets of Linked open data, as basis to create new applications inside the IES Cities platform.

The Query Mapper has been the key tool to integrate the service into the IES Cities platform. By following the design principles of the IES Cities, *Bicicas* services has been easy to integrate within the IES Cities infrastructure, which, in turns, have made the offered services more visible to citizen users. In addition, its use is coherent with other applications already integrated in the platform, and has become a source of new public transport information to both, for citizens as final users of the service and for decision makers, as it has become a

[4] http://www.bicicas.es/estado/EstadoActual.asp

[5] http://www.georss.org

source of issues alerts. Furthermore, the data source created for the application is accessible through the homogenous mechanism provided by IES Cities and, therefore, it can be contributed not only by citizens using the application, but also available to be used by other developers to create their own applications.

Portability evaluation was done by deploying the PhoneGap[6] application on both iOS 5 (simulator) and Android 4.1 (Samsung Galaxy SIII mini), proving though that several of the most recent versions of PhoneGap still have different compatibility issues with both operating systems. In addition, we have carried out a brief evaluation of performance, revealing the transmission time between services and mobile devices to be the bottleneck of the system, but still resulting in execution times below a second.

7. Conclusions and further work

IES Cities is an open platform for publishing data accessible at different sources and provided by public authorities, decision makers, urban planners, public bodies and, of course, citizens in general. One of its main goals is to allow all the participants contributing to the data ecosystem and not to only be passive consumers of the available data. Furthermore, IES Cities envisions fostering the creation of urban services (e.g., mobile applications) that exploit and contribute to the Smarts City's data.

The proposed architecture enforces interoperability of data by providing RESTful APIs for all hosted services, the management of associated entities and the consumption and update of data sources. As the IES Cities infrastructure aims to serve as a starting point for cities that begin to apply open data and envision the creation of a Smart City based on urban services, IES Cities also provides functionality to manage all related entities: applications, data sources, councils and users.

We have shown how the IES Cities platform addresses three important aspects when creating cities that are more inclusive for the citizens. First, the adaptation of heterogeneous and non-structured data from already existing council repositories. The common access mechanism provided by the Query Mapper module of the architecture eases the access to datasets available in multiple formats with a common SQL query mechanism. Second, the platform also focuses on facilitating the creation of urban applications by providing developers with a set of tools to create urban applications.

The IES Cities architecture has also been designed with the idea of serving as a starting point for those public administrations wanting to adopt the open data approach. This goal is achieved thanks to a mechanism that the platform provides to solve the data management cycle and the integration of legacy data sources co-existing in the city ecosystem with those that developers create or newly introduced by the public administration.

Finally, we have validated the proposed architecture in four different European Cities within the consortium and another external city by developing different applications. These applications have been developed as part of the Smart City strategy of each of the participant cities and by other third-party collaborators. This paper summarizes the work done in two of these applications. In addition, the platform has also been validated through the organization of various events aimed to the development of urban applications using the provided data discovery and homogenous access mechanisms. The results have been gathered by interviewing developers, which have provided useful insights about the utility of the proposed approach.

As future work, we plan to continue evaluating the proposed approach with other future application developments and events. Furthermore, and because of the user validation events

[6] http://phonegap.com/

organized during the platform's development, we have detected an aspect that requires to be addressed in order to provide a more integration solution for exploiting the Smart City's data ecosystem: the possibility to federate different data sources allowing consumers to integrate data from different sources. Currently, this process must be carried directly by the consumers accessing to the information, however, due to homogenous data structure provided by IES Cities (i.e., relational view), the platform could provide a more integrated solution to its users.

In addition, we are also planning for future version of the platform to include a mechanism allowing a finer control of the data modifications included by the users. This system will allow administrators to track and accept data changes individually, in a similar way to a version control system for software. This functionality will allow dataset administrators to introduce in their curated datasets those change proposals after validating that they conform to their quality requirements.

Acknowledgments

The European Union's Competitiveness and Innovation Framework Programme has supported this work under grant agreement No. 325097.

References

1. M.F. Goodchild, Citizens as sensors: the world of volunteered geography, GeoJournal 69 (4) (2007) 211-221.
2. M. d'Aquin, J. Davies, E. Motta, Smart Cities' Data: Challenges and Opportunities for Semantic Technologies, IEEE Internet Computing 19 (6) (2015) 66-70.
3. Wikipedia, the free encyclopaedia. http://www.wikipedia.org (accessed on 26/07/2016)
4. OpenStreetMap, http://www.openstreetmap.org/ (accessed on 26/07/2016)
5. FixMyStreet, http://www.fixmystreet.com/ (accessed on 26/07/2016)
6. FixMyTransport, http://www.fixmytransport.com/ (accessed on 26/07/2016)
7. The Apps for Smart Cities Manifesto, http://www.appsforsmartcities.com/?q=manifesto (accessed on 26/07/2016).
8. A. Townsend, R. Maguire, M. Liebhold, M. Crawford, A planet of civic laboratories: The future of cities, information, and inclusion. Institute for the Future Technology (2010), http://iftf.me/public/SR-1352_Rockefeller_Map_reader.pdf (accessed on 26/07/2016).
9. L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787-2805.
10. S. F. Ochoa, R. Santos, Human-centric wireless sensor networks to improve information availability during urban search and rescue activities. Information Fusion 22 (2015) 71-84.
11. A. Monares, S. F. Ochoa, R. Santos, J. Orozco, R. Meseguer, Modelling IoT-based solutions using human-centric wireless sensor networks. Sensors, 14 (9) (2014) 15687-15713.
12. T. Heath, C. Bizer, Linked Data: Evolving the Web into a Global Data Space, Synthesis Lectures on the Semantic Web, First Ed. Morgan & Claypool Publishers (2011).

13. A. Kittur, B. A. Pendleton, B. Suh, T. Mytkowicz, Power of the Few vs. Wisdom of the Crowd: Wikipedia and the Rise of the Bourgeoisie. In: Proceedings of the 25th Annual ACM Conference on Human Factors in Computing Systems (CHI 2007), (2007).
14. J. C. Augusto, A User-Centred Software Development Process, In: Proceedings of the 10th International Conference on Intelligent Environments (2014) 252-255.
15. J. C. Augusto, V. Callaghan, D. Cook, K. Achilles, Intelligent environments: a manifesto, Human-centric Computing and Information Sciences 3 (12) (2013).
16. T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, K. Kim, TinyREST: A protocol for integrating sensor networks into the internet, In: Proceedings of REALWSN, (2005). 101-105.
17. W. Drytkiewicz, I. Radusch, S. Arbanowski, R. Popescu-Zeletin, pREST: a REST-based protocol for pervasive systems, In: Mobile Ad-hoc and Sensor Systems, IEEE International Conference on (2004), 340-348.
18. CKAN – The open source data portal software, http://ckan.org/ (last access 23/11/2015)
19. Open311 – A collaborative model and open standard for civic issue tracking, http://open311.org/ (last access 23/11/2015)
20. D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. IEEE Transactions on Services Computing 3 (3) (2010) 223-235.
21. Socrata – The data platform for 21st century digital government, http://www.socrata.com (accessed on 26/07/2016)
22. Apache Stanbol, https://stanbol.apache.org (accessed on 26/07/2016)
23. D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz. C. Truong, R. Richardson, SPITFIRE: toward a semantic web of things, IEEE Communications Magazine 49 (11) (2011) 40-48.
24. J. Howe, The rise of crowdsourcing. Wired magazine 14 (6) (2006) 1-4.
25. MK:Smart, http://www.mksmart.org (accessed on 26/07/2016)
26. HyperCat Consortium, http://www.hypercat.io (accessed on 26/07/2016)
27. The Data Tank – an open data management systems maintained by OKFN Belgium, http://thedatatank.com/ (accessed on 26/07/2016).
28. FI-WARE, http://www.fiware.org (accessed on 26/07/2016)
29. W3C, PROV Model Primer. Working Group Note 30 April 2013, http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/ (accessed on 26/07/2016).
30. ANTLR – Another tool for language recognition, http://www.antlr.org (accessed on 26/07/2016)
31. E. Prud'Hommeaux, A. Seaborne, SPARQL query language for RDF. W3C recommendation (2008) https://www.w3.org/TR/rdf-sparql-query/ (accessed on 26/07/2016)
32. Waze Mobile. Waze. https://www.waze.com/ (accessed on 26/07/2016).

Unai Aguilera is a researcher and project manager at MORElab "Envisioning Future Internet" Research Group, DeustoTech-INTERNET Research Unit belonging to DeustoTech -- Deusto Institute of Technology. He holds a BSc in Computer Engineering and a PhD in Computer Science from the University of Deusto. He is also a part-time lecturer at the University of Deusto.

Oscar Peña is a PhD Student at the MORElab Research Group at DeustoTech - Deusto Institute of Technology. He received his BSc in Computer Engineering in 2012 and the MSc in Development and Integration of Software Solutions in 2013 from the University of Deusto.

Óscar Belmonte is an associate professor in the Department of Computer Languages and Systems at Universitat Jaume I de Castellón (Spain), where he teaches Programming Agile Software Development and Web Development, particularly in the Master's Degree in Geospatial Technologies. He holds a Bachelor in Physics (1992) and a PhD in Physics (2002) both from the University of Valencia. His current research interests are Indoor Location, Machine Learning and the Internet of Things. He has led some research project in the national and regional areas, and has participated as a researcher in some European research projects.

Diego López-de-Ipiña is an Associate Professor/Reader at the University of Deusto, where he is the Principal Researcher at MORElab "Envisioning Future Internet" Research Group, DeustoTech-INTERNET Research Unit belonging to DeustoTech -- Deusto Institute of Technology. He holds a BSc in Computing from the Faculty of Engineering of the University of Deusto, an MSc in Distributed Information Management Systems from University of Essex and a PhD in Engineering from University of Cambridge.