

Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture

Eduardo Felipe Zambom Santana, University of São Paulo
Ana Paula Chaves, Federal Technological University of Paraná
Marco Aurelio Gerosa, University of São Paulo
Fabio Kon, University of São Paulo
Dejan S. Milojcic, Hewlett Packard Labs Palo Alto

Making cities smarter help improve city services and increase citizens' quality of life. Information and communication technologies (ICT) are fundamental for progressing towards smarter city environments. Smart City software platforms potentially support the development and integration of Smart City applications. However, the ICT community must overcome current significant technological and scientific challenges before these platforms can be widely used. This paper surveys the state-of-the-art in software platforms for Smart Cities. We analyzed 23 projects with respect to the most used enabling technologies, as well as functional and non-functional requirements, classifying them into four categories: Cyber-Physical Systems, Internet of Things, Big Data, and Cloud Computing. Based on these results, we derived a reference architecture to guide the development of next-generation software platforms for Smart Cities. Finally, we enumerated the most frequently cited open research challenges, and discussed future opportunities. This survey gives important references for helping application developers, city managers, system operators, end-users, and Smart City researchers to make project, investment, and research decisions.

General Terms: Digital Cities, Internet of Things, Big Data, Cloud Computing, Cyber-Physical Systems, Middleware, Infrastructure

Additional Key Words and Phrases: Wireless sensor networks, Software platforms, Middleware, Infrastructure

1. INTRODUCTION

Since 2009, most of the world's population lives in cities [United Nations 2009]. Current resources and infrastructure are hardly enough to cope with the increasing demand that population growth and geographic concentration generates [Caragliu et al. 2011]. Making cities smarter can help optimize resource and infrastructure utilization toward increased sustainability. One approach involves creatively combining the large amounts of data generated by multiple city sources (such as sensor networks, traffic systems, user devices, and social networks) to create integrated services and applications, improving city services, and making better use of city resources. However, efficiently and effectively using all these data sources is a challenge.

Initiatives for developing Smart City systems have been proposed in a wide range of city services, such as transportation [Djahel et al. 2014], traffic control [Barba et al. 2012], air pollution [Vakali et al. 2014], waste management [Perera et al. 2014], health care [Hussain et al. 2015], public safety [Galache et al. 2014], water [Pérez-González and Díaz-Díaz 2015], energy [Yamamoto et al. 2014], and emergency management [Asimakopoulou and Bessis 2011]. However, most of these solutions focus on a specific domain, target a specific problem, and were developed from scratch, with little software reuse. They do not interoperate, leading to duplication of work, incompatible solutions, and non-optimized resource use.

Integrating all of these domains into a complete and consistent solution require basic services from the underlying software infrastructure. Such basic services could be provided by a novel, comprehensive software platform, which could include facilities for application development, integration, deployment, and management, easing the

construction of sophisticated Smart Cities applications. We define a software platform for Smart Cities as

“an integrated middleware environment that supports software developers in designing, implementing, deploying, and managing applications for Smart Cities.”

Many challenging issues still need to be addressed before a highly effective software platform for Smart Cities can be created, including: enabling interoperability between a city’s multiple systems, guaranteeing citizens’ privacy, managing large amounts of data, supporting the required scalability, and dealing with a large variety of sensors.

In the research described in this paper, we evaluated initiatives for developing software platforms for Smart Cities, aiming to comprehensively analyze relevant functional and non-functional requirements, according to the literature. Based on the analysis, we derived a reference architecture that addresses these requirements. With this survey, we intend to clarify important aspects of the design, development, and management of Smart Cities platforms. To do so, we examined 23 Smart Cities software platforms, aiming to answer the following **general research question**:

What is required for the development of a software platform for enabling the construction of scalable integrated Smart City applications?

We investigated three more specific research questions:

RQ1: “What are the enabling technologies used in state-of-the-art software platforms for Smart Cities?”

RQ2: “What are the requirements that a software platform for Smart Cities should meet?”

RQ3: “What are the main challenges and open research problems in the development of next generation robust software platforms for Smart Cities?”

To answer research question RQ1, we identified the most common enabling technologies employed in platforms for Smart Cities. As described in Section 2.2, we grouped them into four main categories: **Internet of Things (IoT)** [Atzori et al. 2010], applied to control sensors and actuators responsible for retrieving information from the city; **Big Data** [Mayer-Schönberger and Cukier 2013], to support storage and processing of the data collected from the city; **Cloud Computing** [Armbrust et al. 2010], to provide elasticity to the services and data storage; and **Cyber-Physical Systems** [White et al. 2010], to enable the interaction of systems with the city environment. To answer RQ2, we identified the most common functional and non-functional requirements for developing a platform for Smart Cities, as described in Section 3.3. Finally, to answer RQ3, we explored the main challenges researchers identified in developing software platforms for Smart Cities, as discussed in Section 4.

Combining the results of the three research questions, we derived a reference architecture. This architecture presents components for implementing a software platform for Smart Cities, based on the most common enabling technologies, the requirements, and challenges surveyed in this research. We also discuss the critical implications of platforms for Smart Cities in the Section 6.

The remainder of this paper is organized as follows. Section 2 presents the definition of Smart Cities and introduces the four enabling technologies for platforms for Smart Cities. Section 3 presents the platforms, architectures, and implemented systems for Smart Cities, grouped according to the enabling technologies that each platform uses. Section 4 points out challenges and open research problems in the development of a platform for Smart Cities. In Section 5, we present a reference architecture for software platforms for Smart Cities. In Section 6, we discuss the relationship between the

requirements and the enabling technologies as well as their implications for the development of software platforms for Smart Cities. Section 7 presents the related work and, finally, Section 8 presents our conclusions.

2. MAIN CONCEPTS

We now introduce the main concepts used in the discussions within this survey. We first present definitions of Smart Cities and, then, discuss the most adopted enabling technologies for the development of software platforms for Smart Cities.

2.1. Smart Cities

The term “Smart City” has many different definitions. Some exceed the software context, focusing only on social or business aspects. Regarding software systems, many authors define a Smart City as the integration of social, physical, and IT infrastructure to improve the quality of city services [Caragliu et al. 2011; Hollands 2008]. Other authors focus on a set of Information and Communication Technology (ICT) tools used to create an integrated Smart City environment [Hollands 2008; Washburn et al. 2009; Hall et al. 2000].

Giffinger et al. [Giffinger et al. 2007] assert that a Smart City has six main dimensions: smart economy, smart people, smart governance, smart mobility, smart environment, and smart living. Many authors adopt this definition [Hernández-Muñoz et al. 2011; Papa et al. 2013] and there are even benchmarks to produce a ranking of the smartest city using these dimensions¹.

In their definition of Smart Cities, Washburn et al. [Washburn et al. 2009] and Hall et al. [Hall et al. 2000] emphasize integrating software services and applications to improve regular city services and the lives of their citizens. Following this idea, Kanter and Litow [Kanter and Litow 2009] declare that creating independent software for each city domain is not sufficient for creating an environment for Smart Cities. They contend that all city sub-systems (such as transport, education, energy, and water) must be linked in a network as an organic whole to provide integration among all city subsystems. Caragliu et al. [Caragliu et al. 2011] definition of Smart Cities highlights the significant benefit of sustainability and management of natural resources.

We are aligned with the vision that a city must have an integrated environment to facilitate the interoperability between the city’s sub-systems. Based on that, in our view:

“a Smart City is a city in which its social, business, and technological aspects are supported by Information and Communication Technologies to improve the experience of the citizen within the city. To achieve that, the city provides public and private services that operate in an integrated, affordable, and sustainable way.”

To make a city smarter, it is desirable to integrate services and applications in a unified technological infrastructure. A sensible way to make the above reality is with a well-designed software platform providing the necessary infrastructure for dealing with large volumes of data, a wide variety of devices and applications, system interoperability, and other problems related to Smart City environments.

There are multiple smart city initiatives in a variety of countries around the world, with different maturity levels and applications in different domains. Most of the initiatives are in Europe [Caragliu et al. 2011; Manville et al. 2014], the USA², Japan, and South Korea [Liu and Peng 2013]. Isolated initiatives exist in countries such as Brazil [Fortes et al. 2014] and the United Arab Emirates [Janajreh et al. 2013]. Figure

¹Smarts Cities in Europe - <http://www.smart-cities.eu>

²10 Smartest Cities in USA - <http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america>

1 presents a map with cities that have at least one Smart City project included in this survey. The map shows that most of the projects are located in developed countries, a few in developing countries, and none in underdeveloped countries, where the need for improvements in urban quality of life are most pressing.



Fig. 1. Smart Cities initiatives covered in this survey.

2.2. Enabling Technologies

To answer the question “*What are the main enabling technologies used in state-of-the-art software platforms for Smart Cities?*”, we present the most common enabling technologies that we found in our literature review. We observed four main technologies used by software platforms for Smart Cities: Cyber-Physical Systems, Internet of Things, Big Data, and Cloud Computing. In this section, we give an overview and relate them to Smart City research. These technologies are used later in this paper to group the analyzed platforms and help to understand better the requirements that the platforms must address.

Figure 2 presents an overview of the four enabling technologies that we found in our survey and examples of how they contribute to a platform for Smart Cities.

2.2.1. Cyber-Physical Systems. Cyber-Physical Systems (CPS) can be characterized as the use of computation and communication technologies to improve the features of physical systems. Wan et al. [Wan et al. 2010] define CPS as integration of computation with physical processes. The authors suggest the use of local and remote computational models in networked embedded computers to monitor and control physical processes.

Many real-world applications already leverage CPS [White et al. 2010], such as Smart Cities, power grid control systems, and electronic medical devices. However, some authors [Wan et al. 2010] claim that existing ICT solutions do not support applications with dynamically changing physical contexts. Thus, applying CPS should introduce this requirement to Smart City applications. According to Gurgun et al. [Gurgun et al. 2013], CPS enables applications to become aware of the changes in the physical context adapting their execution according to it.

An example of a Cyber-Physical System related to Smart Cities is WreckWatch [White et al. 2010], an application for detecting traffic accidents. This application was developed for smart phones; it reads the device’s accelerometer and GPS getting the driver’s current speed and acceleration. In case of a strong deceleration, the data is

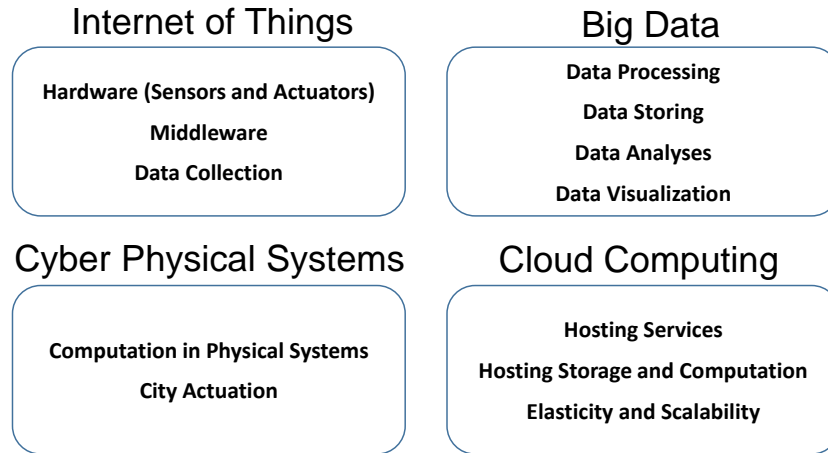


Fig. 2. Platforms for Smart Cities Enabling Technologies

analysed using an accident prediction model and if it indicates an accident, the application generates an alert to a centralized server.

2.2.2. Internet of Things. Coetzee and Eksteen [Coetzee and Eksteen 2011] define IoT as situations where objects become part of the Internet. According to the authors, the objects have to be uniquely identified, with recognized position and status, and accessible to the network. Gubbi et al. (2013) [Gubbi et al. 2013] define three components in an IoT environment: the hardware, which includes sensors, actuators, and embedded communication hardware; a middleware, which processes and stores data received from the hardware; and a presentation layer, in which users access, manipulate, and visualize data extracted from the hardware. In this sense, this is very similar to what we expect from a platform for Smart Cities.

The very large number of devices used to collect data from cities forces platforms for Smart Cities to use IoT technologies. The data collected from these devices must be transmitted via interconnected networks so that they can be grouped and processed to provide advanced Smart City services. Zanella et al. [Zanella et al. 2014] present multiple potential uses of the Internet of Things for Smart Cities, e.g., monitoring the health of historical buildings, detecting the load level of waste containers, sensing noise in central areas of the city, observing the conditions of traffic lights, and analyzing the usage of energy in Smart Homes.

2.2.3. Big Data. Most authors consider Big Data as a set of techniques and tools to store and manipulate large data sets whereas conventional technologies, such as relational databases and sequential processing tools, cannot deal with such a vast volume of data. There are four major characteristics of Big Data [Chen et al. 2014; Demchenko et al. 2014]:

- **Volume:** the scale of data generated and collected is rapidly increasing, and Big Data tools must deal with this challenge. In Smart Cities, the volume of data will be massive, coming from many data sources distributed across the city.
- **Variety:** data is collected from different sources, and have structured, semi-structured, or unstructured formats, such as video records, relational databases, and

- raw texts, respectively. This is important for Smart Cities, since city data is collected from multiple sources, such as surveillance cameras, sensors, and citizen devices.
- **Velocity:** data processing must be fast and, in some cases, real-time, or it may be useless. City infrastructure, operators, and managers need to respond to urban problems, such as traffic jams, accidents, and floods, in short time.
 - **Veracity:** because of the large amount of data collected, and the use of multiple data sources, it is important to ensure data quality, because errors in the data or the usage of unreliable sources can compromise its analysis. In cities, incorrect GPS readings, malfunctioning sensors, and malicious users can be sources of poor data.

Figure 3 relates the four Vs of Big Data with Smart Cities' needs.

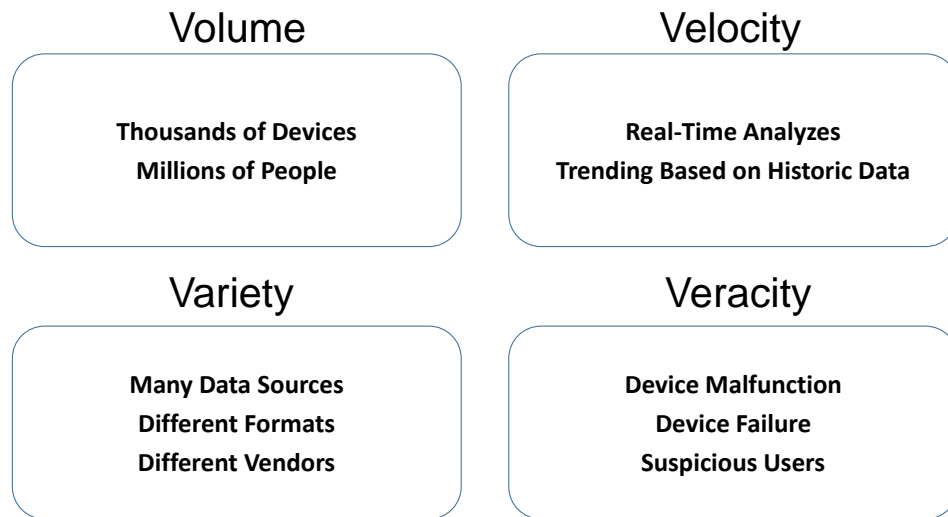


Fig. 3. 4 Vs of Big Data

Smart Cities already use Big Data tools to support the amount of data generated from city devices. Sensor networks regularly transmit data about city conditions, such as temperature, air quality, and pluviometry. Citizens generate data using smartphones and social networks, and vehicles continuously send their positions.

Big Data tools are already used by Smart City platforms, including: NoSQL databases [Khan et al. 2013; Bain 2014], such as MongoDB and HBase; parallel data processing tools [Parkavi and Vetrivelan 2013; Takahashi et al. 2012], such as Apache Hadoop and Apache Spark; real-time data streams processing tools [Girtelschmid et al. 2013], such as Apache Storm; and visualization tools [Khan et al. 2013], such as Rapid-Miner.

Al Nuaimi et al. [Al Nuaimi et al. 2015] discuss potential applications of Big Data tools in Smart Cities, such as recognizing traffic patterns and using historic data to locate the causes and avoid traffic jams, facilitating the decisions of city governments using analyses of large data sets, and predicting the use of resources, such as electricity, water, and gas, in different situations using historic and real-time data.

2.2.4. Cloud Computing. Cloud Computing offers a very large, elastic, and highly available infrastructure for both data storage and computation, which is essential for complex Smart City systems. In addition, a Smart City environment can be highly dynamic, requiring reconfigurations of the underlying infrastructure, which is also supported by Cloud Computing.

Many authors, such as [Distefano et al. 2012; Aazam et al. 2014], have advocated combining IoT and Cloud Computing, coining the term “Cloud of Things”. Their idea is to store and process all the data from an IoT network in a cloud computing environment, which is currently used in some Smart City projects [Mitton et al. 2012; Tei and Gurgun 2014].

Another concept related to a cloud computing environment in Smart Cities is Software as a Service (SaaS), which provides the sensor data with a cloud computing infrastructure. The work of Perera et al. [Perera et al. 2014] extends this concept, using the term “Sensing as a Service”. The ClouT platform, presented in [Tei and Gurgun 2014], also uses the concept of software services and defines the terms City Application Software as a Service (CSaaS) and City Platform as a Service (CPaaS).

Some authors relate the use of Cloud Computing, Big Data, and IoT [Chen et al. 2014; Aazam et al. 2014], because a cloud environment is an ideal infrastructure to store data and execute services. Hence, the data generated from an IoT middleware can be stored and processed in a cloud environment using Big Data tools. This synergistic combination helps to support important non-functional requirements such as scalability, elasticity, and security.

3. PLATFORMS FOR SMART CITIES

We describe here various platforms for Smart Cities presented in the literature. All platforms use at least one of the enabling technologies discussed in Section 2.2.

To find these studies, we used the following query string: (“Smart City” or “Smart Cities”) and (Platform or Middleware or Architecture). After analyzing the query results, we focused our study on 47 papers describing Smart City platforms and applications. We did not include in our search other terms that are more rarely used to describe the application of ICT in cities, such as “Knowledge City”, “Intelligent City”, and “Connected City”. Figure 4 illustrates the use of these expressions in recent years using Google Trends.

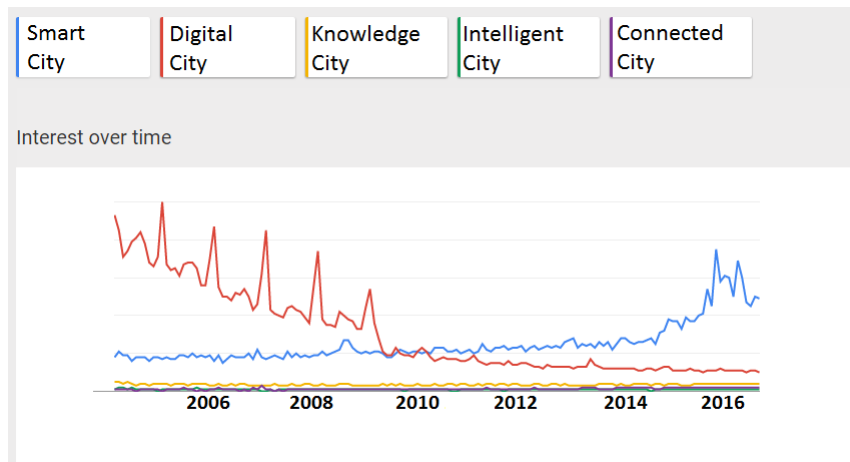


Fig. 4. Trends in Smart City related terms

Since the expression “Digital City” is still used, we analyzed the definition of this expression and the differences with “Smart Cities”. We found that, normally, the description of a digital city relates to the use of digital technologies in a city, but not with the goal of making smart services and improving the city’s overall infrastructure. In a digital city, the integration of the multiple systems is not at stake. The differences between these two concepts are discussed by Cocchia [Cocchia 2014] and by Yin et al. [Yin et al. 2015].

The next subsection describes existing platforms, developed as research projects with different approaches. Subsection 3.2 shows systems developed using these platforms. Finally, in Subsection 3.3, we present a set of functional and non-functional requirements extracted from our analysis of the platforms and systems described in the previous subsections.

3.1. Platform Categories

To facilitate the presentation, we divided the platforms into five categories, according to the enabling technologies that each platform uses. Figure 5 presents an overview of the platforms for Smart Cities that we analyzed. In this figure, we can observe that most platforms use Cloud Computing. Almost all of them use at least one more enabling technology, more commonly IoT and Big Data.

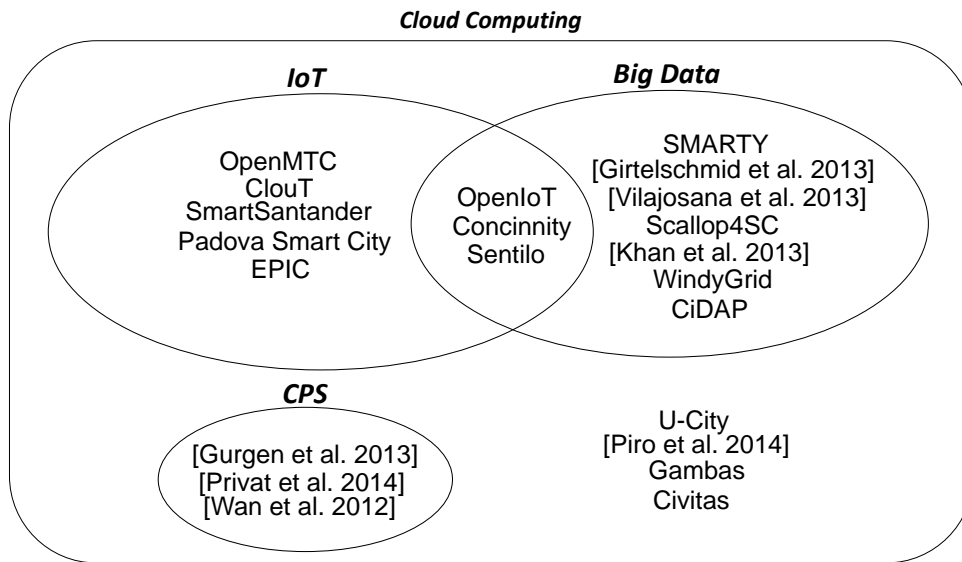


Fig. 5. Use of Enabling Technologies by Smart City Platforms

3.1.1. Internet of Things and Cloud Computing. In this section, we present the platforms that use both IoT and Cloud Computing as enabling technologies.

SmartSantander is an experimental infrastructure to support the development and deployment of Smart City applications and services [Sanchez et al. 2014]. The project is centered in Santander, Spain, with smaller facilities in other European cities. The platform processes a large variety of information, including data about traffic conditions, temperature, CO_2 emissions, humidity, and luminosity. Currently, the project has implanted more than 20,000 sensors in the city.

Padova Smart City [Zanella et al. 2014] uses IoT to create a sensor network in the city of Padova, Italy. Using more than three hundred sensors, the platform collects environmental data, such as CO2 emissions and air temperature, and monitors street lights. A feature highlighted in this platform is the use of common protocols and data formats to allow interoperability among multiple city systems.

The European Platform for Intelligent Cities (EPIC) project [Ballon et al. 2011] proposes a complete IoT Middleware to facilitate the use and management of the Wireless Sensor Network (WSN). This middleware aims to deal with the heterogeneity, interoperability, scalability, extensibility, and configurability problems in a WSN.

ClouT [Tei and Gurgen 2014] proposes a two-layer architecture to collect data from the WSN and manage the sensors and actuators in the city network [Galache et al. 2014]. The first layer is the Sensors and Actuators Layer, which handles data from the WSN. The second layer, the IoT Kernel Layer, manages and monitors the sensors and actuators network.

OpenMTC [Elmangoush et al. 2013] (Open Machine Type Communications) is a Machine-To-Machine (M2M) based communication platform for Smart Cities. Its goal is to enable efficient communication among a large number of devices, associating them with multiple services. To achieve this, the platform supports standard interfaces to various types of devices, data/event processing methods to achieve real-time performance, and easy application development, providing a software development kit.

The analysis of the platforms aforementioned led to the identification of four major functional requirements: management of a WSN, management of the data collected from the city, management of services and applications, and an infrastructure to make the data from the platform available to city applications. This analysis also led to the identification of five non-functional requirements: adaptation, interoperability, scalability, extensibility, and configurability.

We identify two weak points of these platforms: (1) the lack of pre-processing components to verify the integrity of the data collected from the city and make small transformations of the data, such as aggregations, and (2) most of the platforms do not include a discussion about security concerns.

3.1.2. Internet of Things, Cloud Computing, and Big Data. In this subsection, we present platforms that use IoT, Cloud Computing, and Big Data as enabling technologies.

OpenIoT³ is an open source middleware for the development of IoT-based applications. It has an API to manage the WSN, and a directory service to dynamically discover the sensors deployed in the city; it also has a layer for service definition and access. Big Data tools are used to store and analyze the data from the platform. A Smart City project called Vital [Petrolo et al. 2014] builds on this platform and uses the term "Cloud of Things" to refer to the use of Cloud Computing and IoT.

The Concinnity project provides a platform for managing data and applications following the PaaS model [Wu et al. 2014], with which its authors built Big Sensor Data Applications. However, this platform focuses on multiple data sources such as the WSN, social networks, and data from platform users. It also includes a service directory where developers can find and publish services facilitating its reuse.

OpenIoT and Concinnity, offer developers tools to implement applications directly on the platform. OpenIoT allows the mash-up of the services defined in the platform and automatically creates a visual interface for end-users. Concinnity provides a set of development tools, such as a Workflow Editor and Engine, a Service Publisher, and an Application Editor.

³OpenIoT - <https://github.com/OpenIoTOrg/openiot>

Sentilo [Bain 2014] is platform that deals with the management of sensors and actuators, designed for Smart Cities that looks for openness and interoperability. Sentilo uses IoT concepts to control the WSN, and Cloud Computing to share data with the applications. Big Data tools are mainly used to collect and store data from sensors, ensuring platform scalability. The Sentilo project was originally designed to be deployed in the city of Barcelona; after its deployment, the City released the code under the LGPL and EUPL open source licenses.

The main functional requirements identified for this group of platforms were: management of a WSN, management of data life cycle (collect, store, process), making the data from the platform publicly available, a service directory for application developers, and tools for application development. As non-functional requirements, we identified: interoperability and scalability.

A weak point of these platforms is the lack of streams processing tools to analyze real-time data from the city, an important requirement for many Smart City applications. Another problem is that most of the platforms do not support the customization of services with citizen data. In spite of the privacy problems, offering context-aware, customized services to the citizens is highly desirable.

3.1.3. Cloud Computing and Big Data. In this subsection, we present platforms that use Cloud Computing and Big Data as enabling technologies.

Vilajosana et al. [Vilajosana et al. 2013] present a platform for Smart Cities based on Cloud Computing and Big Data, whose main components are data management and service hosting. It includes an Open Data API allowing third-party applications to access the data stored on the platform. Big Data tools are used to collect data streams and analyze data, such as prediction and inference.

Scallop4SC (SCALable LOGging Platform for Smart City) [Takahashi et al. 2012; Yamamoto et al. 2014] uses Big Data to process a large volume of data gathered from smart buildings. The platform uses information about the building, such as water and energy consumption, temperature, air humidity, and the amount of garbage generated. Periodically, the buildings send data to the platform for processing. The objective is to analyze smart building data, for which it uses the MapReduce algorithm.

CiDAP [Cheng et al. 2015] is a big data analytics platform deployed into the Smart-Santander testbed. The platform uses data collected from SmartSantander and analyzes it to understand the behavior of the city. The main components of this platform are: the agents, which collect data from the SmartSantander platform; the Big Data repository for storing the data; the Big Data processing for intensive data processing and analytics; and a CityModel server, responsible for interfacing with external applications. This platform uses Apache Spark [Zaharia et al. 2010] to process the data.

[Khan et al. 2015] propose a Smart City architecture based on Big Data to achieve the necessary availability and scalability required for a Smart Cities platform. The architecture has three layers: one to collect, analyze, and filter data; another to map and aggregate data to make it semantically relevant; and a third layer where users can browse and recover the data processed from the other two layers. The implementation of the architecture uses only open source projects, and the authors have presented tools for all layers [Khan et al. 2013].

WindyGrid [Thornton 2013], an initiative of the City of Chicago, is a platform for Smart Cities, whose objective is to present real-time and historical data with a unified view of city operations. Big Data technologies, such as the MongoDB NoSQL database and parallel data processors, were used to develop the platform.

SMARTY [Anastasi et al. 2013] is a project aimed at providing tools and services for mobility and flexible city transport systems. Its software platform collects data from multiple sources, such as traffic flow, user location, transport service delays, and

parking availability. A network of low-cost sensors collects data from the city and social networks are continuously monitored to get data from citizens. The platform processes the massive amount of data generated by the city with data mining techniques, such as classification, regression, and clustering.

The platform proposed by Girtelschmid et al. [Girtelschmid et al. 2013] uses semantic technologies to create a platform for Smart Cities, adding flexibility in system configuration and adaptation. However, to overcome the performance bottlenecks normally associated with ontology repositories and reasoning tools, the authors combine their semantic techniques with Big Data processing methods.

The main functional requirements identified for this group of platforms were: data management, such as collecting, analyzing, and visualizing data; large scale data processing, such as batch and real-time processing; and the use of semantic techniques combined with Big Data. As non-functional requirements, we identified scalability and adaptation.

Most of the platforms in this section do not have an IoT layer, and do not indicate how the data is collected from the city; the exception is CiDAP, which uses the SmartSantander testbed as an IoT middleware. Another drawback is that most of the platforms do not include a discussion about security concerns.

3.1.4. Cloud Computing. In this subsection, we present platforms that use only Cloud Computing as an enabling technology.

Piro et al. [Piro et al. 2014] present a two-layered service platform for the creation of Smart City applications. The first is a low-level layer that controls the communication among the city WSN devices. The second layer collects the data from the devices and provides services for the development of applications that use the data from the city.

U-City [Lee and Rho 2010] is a platform for the creation of smart ubiquitous cities. The platform offers several service management features, such as autonomic service discovery, service deployment, and context-aware service execution. It also offers pre-defined services such as an inference engine, a context-aware data service, and a portal for the management of the platform.

Gambas, a middleware for the development of Smart City applications [Apolinarski et al. 2014], supports data acquisition, distribution, and integration. The platform also provides an application runtime to facilitate the development and deployment of services using city data and a service registry. The middleware supports context-awareness, so that Smart City services can adapt to the citizen situation, behavior, and intent. All communication in the platform is encrypted to ensure citizen's privacy and security.

Civitas [Villanueva et al. 2013] is a middleware to support the development of Smart Cities services. It is used to facilitate the development and deployment of Smart City applications, and to avoid the emergence of "information islands" [Qiu et al. 2010], i.e., disconnected applications that do not share relevant information. Citizens connect to the middleware via a special device called the Civitas Plug, which ensures the privacy and the security. The middleware has two main design principles to facilitate the application integration: *Everything is a Software Object*, which promotes the consistency of the software design and reusability of the middleware; and *Independence of the City Layout*, meaning that city services should not work with just one city layout.

The main functional requirements identified for this group of platforms were: service management and data management. As non-functional requirements, we identified: security, privacy, and context awareness.

A drawback of the platforms presented in this section is that none of them use known frameworks to implement components, such as the inference engine and processing tools, which might make difficult the maintenance of the platform. Another problem is

that the platforms do not describe a mechanism to allow external access to the platform data.

3.1.5. Cloud Computing and Cyber-Physical Systems. In this section, we present platforms that use Cloud Computing and Cyber-Physical Systems (CPS) as enabling technologies.

Gurgen et al. [Gurgen et al. 2013] present a middleware for Smart Cities automatic services, which includes many “self-” properties, such as self-organization, self-optimization, self-configuration, self-protection, self-healing, self-discovery, and self-description. They justify using cloud computing to provide scalability, reliability, and elasticity to the platform. This platform provides application developers with the contexts of individual users and the city.

Privat et al. [Privat et al. 2014] propose another CPS-based platform, whose main characteristic is self-configuration and self-adaptation capabilities in smart environments, including Smart Cities. This platform provides a shared distributed software infrastructure that collects data and reacts to changes in the environment.

Wan et al. [Wan et al. 2012] propose an event-based CPS platform, which uses an event manager to manage and generate cooperation among M2M components. This platform provides data and services to third-party applications through a publish/subscribe module. The platform also enables the design of event processing flows to manage the mission-critical wireless messages.

The main functional requirements identified for this group of platforms were: automatic reaction to changes in the city environment, communication among city devices, and a publish/subscribe mechanism for applications to communicate with the platform. As non-functional requirements, we identified: configurability, adaptation, and context awareness.

The platforms of this section focus on the deployment, configuration, and execution of CPS devices in the city, but they lack important requirements, such as the monitoring and publication of the data from the devices. They also do not describe any mechanism to verify the data collected from the city, discarding inconsistencies.

3.2. Systems

In this subsection, we illustrate applications built on top of the platforms presented in the previous section. Table I presents an overview of the domains of the analyzed systems.

The GAMBAS middleware was used to develop two applications for the public transportation system in Madrid, Spain. Foell et al. [Foell et al. 2014] present a context-aware urban bus navigator to help travelers find the best buses for their trips. Handte et al. [Handte et al. 2014] describe a system that estimates the number of passengers of city buses using smartphone sensing.

SEN2SOC [Vakali et al. 2014] is a system deployed on the SmartSantander platform that uses data streams from the city (e.g., sensor data) and social networks (e.g., Twitter) to create Smart City applications. Two examples of applications are: capturing the emotional state of city inhabitants, and visualizing the air pollution in the city. Also in SmartSantander, Vlahogianni et al. [Vlahogianni et al. 2014] present an application to predict the utilization of city parking lots.

Two applications were developed using the Padova Smart City platform. Bui and Zorzi [Bui and Zorzi 2011] present a health care system whose main features are monitoring conditions of patients, sending their data directly to doctors, and calling emergency services if the patient has an urgent problem. Bressan et al. [Bressan et al. 2010] present a monitoring application to manage and collect data from all the light posts connected to the platform.

Table I. Domains of Smart City Systems

	City Sensing	Traffic Control	Air Pollution	City Dashboard	Health Care	Safety	Disaster Prevention	Energy Management	Waste Management
GAMBAS		X							
SmartSantander	X	X	X						
Padova Smart City	X			X					
OpenIoT	X								X
WindyGrid	X	X	X		X	X			
ClouT	X	X			X	X	X		
Scallop4SC								X	
Number of Instances	5	4	2	1	2	2	1	1	1

Anagnostopoulos et al. [Anagnostopoulos et al. 2015] present a waste management system implemented using the OpenIoT platform. It presents four models to prioritize critical trash bins, such as bins close to schools, hospitals, and gas stations. The system was used to compare the four models according to the amount of CO₂ emitted and the distance traveled by trash trucks.

The WindyGrid platform [Rutkin 2014] provides three main systems to the city of Chicago: Situational Awareness and Incident Monitoring, to monitor and act on problems that are occurring in the city; Historical Data Analyses, to predict the behavior of city systems, such as traffic and health care; and Advanced Real-Time Analytics, to analyze the current situation of city systems. Some examples of the data used in these systems are: log of emergency (911) calls, traffic conditions, public buildings information, and surveillance cameras.

Galache et al. [Galache et al. 2014] present four systems developed using the ClouT platform: an alert service to warn citizens about earthquakes in Fujisawa, Japan; a civil protection system, which warns the population about environmental risks such as storms and earthquakes in Genova, Italy; a system to help elderly people find healthy activities in the city of Mitaka, Japan; and a sensing application to notify people about events in Santander, Spain, such as cultural acts and traffic accidents.

Yamamoto et al. [Yamamoto et al. 2014] present two systems developed for the Scallop4SC platform, both in the energy management domain. The first system offers a tool for the visualization of household energy consumption, which analyzes data at different levels, such as state, city, and neighborhood. The other system is a wasteful energy detection service that is available for smart homes.

The analyzed applications show that the most explored domains in the literature are traffic, with applications to monitor the streets or help citizens to use the public transport, and city sensing, capturing data from the city using sensors such as air pollution and temperature. Most of the applications are developed external to the platform, using only one or more platform services.

3.3. Requirements for Smart City Software Platforms

To answer the second research question “*What are the requirements that a software platform for Smart Cities should meet?*”, in this section we analyze the functional and non-functional requirements extracted from the analyzed platforms.

We assume that a platform implements a requirement if the literature describing it explicitly states so, or if the platform has a component or module that clearly fulfills that requirement.

3.3.1. Functional Requirements. The main goal of a platform for Smart Cities is to facilitate the development of Smart City applications. Towards this aim, most of the analyzed platforms implement requirements for collecting data from the city, managing and sharing data, and providing tools to facilitate the development of Smart City applications. Table II presents an overview of the functional requirements for Smart City platforms, which we describe in the following:

Table II. Functional requirements for Smart City platforms

	Data Management	Application Run-time	WSN Management	Data Processing	External Data Access	Service Management	Software Engineering Tools	Definition of a City Model
SmartSantander	X	X	X		X			
OpenIoT	X	X	X	X		X	X	
Concinnity	X	X		X	X	X	X	
Civitas	X			X		X		
Gambas	X	X			X	X		X
[Khan et al. 2013]	X			X	X	X	X	
[Girtelschmid et al. 2013]				X	X			
Scallop4SC	X			X	X			
OpenMTC					X	X	X	
[Wan et al. 2012]	X			X		X		
[Piro et al. 2014]						X		
[Gurgen et al. 2013]	X	X	X	X	X	X		
[Vilajosana et al. 2013]	X		X	X	X	X		
ClouT	X	X	X		X			
Padova Smart City	X		X	X	X			
U-City	X	X		X	X	X		
Sentilo	X		X		X			
WindyGrid	X					X		
EPIC	X		X		X	X		
[Privat et al. 2014]								X
SMARTY	X		X	X	X	X		
CiDAP	X			X	X			X
Number of Instances	18	7	9	13	16	14	4	3

- **Data Management:** This is a requirement implemented by most of the platforms for Smart Cities, which includes collection, storage, analysis, and visualization of city data. The analyzed platforms use different techniques for this requirement, such as relational databases [Hernandez-Muñoz et al. 2011; Lee and Rho 2010], big data

tools [Thornton 2013; Cheng et al. 2015], and customized tools implemented by the platform development team [Wu et al. 2014].

- **Applications Run-time:** Some platforms focus on managing the execution of their applications. The goal is to facilitate the deployment and integration of such applications. Some platforms provide a complete environment for developers to deploy their applications [Apolinarski et al. 2014]; others offer an execution run-time service for applications developed with tools the platform provides [Petrolo et al. 2014; Wu et al. 2014].
- **WSN Management:** Many of the analyzed platforms have a Wireless Sensor Network (WSN) management layer to control and monitor the devices deployed in the city. Most of these platforms use IoT concepts to organize and manage the WSN [Hernandez-Muñoz et al. 2011; Tei and Gurgun 2014]. Other platforms [Bain 2014] do not explicitly mention this, but indeed have a software layer to manage the city network devices. Some platforms include features to manage all the device activities, such as adding, removing, and monitoring the sensors and actuators. Two platforms describe a WSN deployed in a city: Padova Smart City [Zanella et al. 2014], with 3000 sensors, and SmartSantander [Sanchez et al. 2014], with more than 20000 sensors.
- **Data Processing:** Some platforms use specific processing components, such as inference engines [Lee and Rho 2010], workflow processing [Wu et al. 2014], and big data processing tools [Takahashi et al. 2012]. These components process large data sets, and their main purpose is to analyze, verify, aggregate, and filter the data from the city. In addition, some platforms [Girtelschmid et al. 2013; Cheng et al. 2015] make real-time analyses of data streams.
- **External Data Access:** Almost all platforms describe an interface for external applications to access the platform data. The most common approach is an API to allow access to the data generated in the city. Some platforms use REST [Hernandez-Muñoz et al. 2011; Elmangoush et al. 2013], others use cloud computing concepts to provide the city data as a service [Ballon et al. 2011], and one proposes an open data platform [Zanella et al. 2014]. Also, a platform [Gurgun et al. 2013] uses the publish/subscribe paradigm to make the data and services available to applications.
- **Service Management:** Most of the analyzed platforms adopt a Service-Oriented Architecture, in which the platform functionalities are offered by services [Issarny et al. 2011]. Some of them use services to provide features to applications, such as access to raw sensors data [Petrolo et al. 2014] and analyzed data [Zanella et al. 2014], and workflow engines [Wu et al. 2014]. Others enable developers to deploy services on the platform and make them available to other applications [Apolinarski et al. 2014; Piro et al. 2014]. Some platforms also use service compositions and choreographies [Issarny et al. 2011] to create new services or applications [Lee and Rho 2010; Piro et al. 2014].
- **Software Engineering Tools:** Some platforms provide a set of tools for the development and maintenance of services and applications. For describing and implementing applications, some platforms create visual interfaces [Petrolo et al. 2014]. Other platforms provide workflow design tools [Wu et al. 2014] to define data or service flows and create Smart City applications. Moreover, some platforms [Khan et al. 2013] use analytics and reporting tools to facilitate the development of data visualization and reports, and two platforms describe the use of a Smart City application SDK [Elmangoush et al. 2013; Apolinarski et al. 2014].
- **Definition of a City Model:** Some platforms provide a model of the city to facilitate the manipulation and understanding of the platform data, and to facilitate the integration of the collected data. For example, in Cheng et al. [Cheng et al. 2015], the city model is used to allow queries in the data from the city sensor network. Privat et al. [Privat et al. 2014] use a finite-state model to represent the possible city data flows.

Based on the functional requirements aforementioned, we can observe that the main platforms activities aim to control the city data life cycle: (1) Collecting the data with a WSN, (2) Managing the data in the platform, (3) Processing the data using city models, and (4) Sharing the raw and processed data allowing external access. These activities are highly related to the enabling technologies, such as IoT with the WSN management, Data Management and Processing with Big Data, and Service Management with Cloud Computing.

3.3.2. Non-Functional Requirements. Most of the non-functional requirements of Smart City platforms are related to large, heterogeneous distributed systems, such as scalability, adaptation, and interoperability. Other non-functional requirements are related to the manipulation of critical and personal data from citizens, such as security and privacy. Table III presents an overview of the non-functional requirements for Smart City platforms, which we describe in the following.

Table III. Non-Functional requirements for Smart City platforms

	Interoperability	Scalability	Security	Privacy	Context Awareness	Adaptation	Extensibility	Configurability
SmartSantander		X	X	X				
OpenIoT	X		X					X
Concinnity	X		X	X				X
Civitas	X		X	X				
Gambas			X	X	X			
[Khan et al. 2013]		X			X		X	
[Girtelschmid et al. 2013]	X	X			X	X		
Scallop4SC		X					X	
OpenMTC	X							
[Wan et al. 2012]	X					X		X
[Piro et al. 2014]			X					
[Gurgen et al. 2013]	X				X	X		
[Vilajosana et al. 2013]	X							
ClouT	X		X					
Padova Smart City	X	X						
U-City			X					X
Sentilo		X	X				X	
WindyGrid			X	X				
EPIC	X	X						
[Privat et al. 2014]					X	X		X
SMARTY	X				X			
CiDAP	X	X			X	X	X	
Number of Instances	13	8	10	5	7	5	4	5

— **Interoperability:** Different devices, systems, applications, and platforms compose a Smart City environment, and all these components must operate in an integrated

- fashion; for example, sensors from multiple vendors, systems implemented in different languages, platforms that share data and users, and legacy systems that have to communicate with the new platforms. Previous work in the field adopted several techniques to handle this requirement: interoperable objects [Villanueva et al. 2013], adopting generic and standard interfaces [Gurgen et al. 2013], applying Semantic Web to integrate all platform components [Girtelschmid et al. 2013], and using a naming mechanism [Cheng et al. 2015] to recognize different devices or data sources.
- **Scalability:** The amount of users, data, and services of a Smart City platform will be massive, and can increase over time. For example, in the SmartSantander testbed, there were more than 20,000 sensors, in a city of 178,000 inhabitants collecting a large amount of city data [Sanchez et al. 2014]; CiDAP collected more than 50 GBs of data in three months [Cheng et al. 2015]. This non-functional requirement is relevant to many functional requirements, such as WSN management [Ballon et al. 2011], data management [Takahashi et al. 2012], and service management [Bain 2014].
 - **Security:** Malicious users can make fraudulent use of services and data provided by the platform. Many platforms have a component or describe mechanisms to handle security, avoiding attacks to the city infrastructure and information theft [Piro et al. 2014; Hernandez-Muñoz et al. 2011; Petrolo et al. 2014].
 - **Privacy:** A Smart City platform collects and manipulates several citizen-sensitive data, such as medical records, user localization, and consuming habits. The challenge is to use these data while hiding, or to avoid saving identifiable information. Some of the strategies used to achieve this requirement are cryptography [Apolinarski et al. 2014], tokens to control the access to the data that users can manipulate [Villanueva et al. 2013], and anonymization [Mylonas et al. 2015].
 - **Context Awareness:** As the city and user situation can change over time, many applications and services can provide better results using contextual information. Some platforms use information from users [Apolinarski et al. 2014; Privat et al. 2014], such as location, activity, and language. Other platforms use information from the city [Khan et al. 2013; Cheng et al. 2015], such as traffic conditions, climate, and air quality. Examples of context use are: displaying a different language in an application to a tourist, and changing the route of a user avoiding polluted areas.
 - **Adaptation:** Related to context awareness, many platforms adapt their behavior based on context in order to achieve fault-tolerance, choose a closer server to improve efficiency, decide for batch or real-time processing, and adapt data from multiple data sources. This requirement is most used in platforms that use CPS as enabling technology [Privat et al. 2014; Wan et al. 2012], but other concepts are used to meet this requirement as well, such as semantic technologies [Girtelschmid et al. 2013].
 - **Extensibility:** The capability to add services, components, and applications to the platform is important to assure that it meets evolving system requirements and user needs. Munoz et al. [Hernandez-Muñoz et al. 2011] state that easy extensibility is valuable because one cannot know what services a city will need. Scallop4SC [Takahashi et al. 2012] uses materialized views that developers extend to implement their applications. Some platforms [Khan et al. 2013; Bain 2014] employ only open source tools, facilitating the platform's extensibility. CiDAP [Cheng et al. 2015] offers extensibility to enable the use of the platform in cities of different scales.
 - **Configurability:** A Smart City platform has many configuration options and parameters that define its behavior at execution time, such as defining pollution and congestion thresholds and the priority of services. Thus, it is important to allow (re)configuration of the many variables of the platform. Two platforms [Wan et al. 2012; Privat et al. 2014] highlighted the importance of self-configurability capacities, because of the huge amount of configurations needed in a Smart City platform. Other

platforms [Lee and Rho 2010; Kim and Lee 2014] provide a portal to centralize the configurations.

Based on the non-functional requirements aforementioned, we can observe that some of them are very important to many functional requirements: such as Scalability, which is valuable to the WSN and Data Management; Security and Privacy, which are important to all data requirements; Extensibility, which is required to the Service Management; and Configurability, which is important to all the functional requirements. The non-functional requirements are very similar to the challenges and open research problems that we present in the next section.

4. CHALLENGES AND OPEN RESEARCH PROBLEMS

To answer RQ3 (“*What are the main challenges and open research problems in the development of next generation, robust software platforms for Smart Cities?*”), we analyzed the challenges pointed out by Smart City research papers. Table IV presents an overview of the main challenges, which we describe in the following.

Table IV. Overview of most cited Challenges and Open Research Problems

Challenge	Description	Technologies/Tools
Privacy	Protecting data collected from citizens, city, and enterprises.	Cryptography, Anonymization, and Access Tokens
Data Management	Managing all the data collected in the platform	NoSQL and Relational Databases and processing tools.
Heterogeneity	Ensuring the interoperability of devices and applications	Standards, Ontology, and a City Unified Model.
Energy Management	Managing the electricity used by devices deployed in the city.	
Communication	Enabling communication among heterogeneous devices.	M2M techniques.
Scalability	Allowing the growth of devices and users connected to the platform.	Distributed tools and algorithms and P2P applications.
Security	Protecting the city data, services, and infrastructure.	Cryptography, Access Tokens and Devices.
Lack of Testbeds	There are not sufficient testbeds to experiment Smart City solutions.	Simulators.
City Models	Defining a model describing the city.	Semantic Web and Ontologies.
Platform Maintenance	Maintaining the city systems and infrastructure.	Monitoring and Alert tools.

- **Privacy:** is the most cited challenge to implementing a Smart City platform; the main reason pointed by Hassan et al. [Hassan et al. 2014] and Balakrishna [Balakrishna 2012] is that the data collected from the city includes personal, enterprise, and governmental data that should not be accessed by other unauthorized users. Wan et al. [Wan et al. 2012] discuss legal problems in using data belonging to platform users.
- **Data Management:** Many authors also cite data management as a challenge, because the platform has to store and process a large amount of data and use efficient and scalable data storage and processing algorithms [Su et al. 2011; Djahel et al. 2014; Perera et al. 2014]. Data Analysis is also a challenge, because it is hard to extract useful knowledge [Hassan et al. 2014]. Another challenge is data trustworthiness; for example, Wu et al. [Wu et al. 2014] claim that a large number of data sources make it difficult to ensure that all the data are correct.

- **Heterogeneity:** This is a challenge because of the differences between the devices in a Smart City, and the difficulty of relating data from different sources [Wu et al. 2014; Su et al. 2011; Wan et al. 2012]. Naphade et al. [Naphade et al. 2011] raise the problem of managing data across all city systems because of variations in data from different sources. Other authors [Wenge et al. 2014] state that a Smart City platform has to define standards across heterogeneous devices, systems, and domains.
- **Energy Management:** Some authors cite Energy Consumption as a challenge to be faced by all the components of the platform, such as sensors, actuators, and servers [Perera et al. 2014]. Moreover, Hassan et al. [Hassan et al. 2014] point out that energy management in a Smart City health care application is also important, because applications or services in domains like this cannot fail due to power outages.
- **Communication:** Since the smart cities of the future will incorporate a massive amount of devices, enabling communication among these devices will be a challenge. Some authors [Wan et al. 2012; Hassan et al. 2014] discuss the domains in a Smart City that depend on mission-critical communication to ensure reliability, such as health care and public safety. In addition, Djahel et al. [Djahel et al. 2014] explain that good communication mechanisms are required to share platform data with applications.
- **Scalability:** Within the next decades, the number of connected devices in a Smart City will continually increase [Balakrishna 2012], requiring a strong level of scalability in the associated software platform. Moreover, the number of users, services, and data stored will increase with population growth and on special events in the city. Su et al. [Su et al. 2011] discuss how a Smart City platform must support large-scale, efficient services. As an example, Sinaeepourfard et al. [Sinaeepourfard et al. 2016] estimated that the city of Barcelona will need more than 1 million sensors to cover all the city, generating more than 8 GB of data every day.
- **Security:** Unauthorized users accessing city services without permission may cause a lot of harm. Hancke et al. [Hancke et al. 2012] consider whether city networks will be safe from cyber-terrorism and cyber-vandalism. Gurgun et al. [Gurgun et al. 2013] highlighted the importance of security in CPS platforms, as such systems control part of the city infrastructure, which a malicious user can corrupt, e.g., by tampering with traffic lights and light posts.
- **Lack of Testbed:** The lack of testbeds is cited by Elmangoush et al. [Elmangoush et al. 2013] and Muoz et al. [Hernandez-Muñoz et al. 2011] as a challenge to the development of platforms for Smart Cities. Without testbeds, it is hard to perform tests and experimentation to discover the real challenges that deploying a Smart City platform will present. Smart City Simulators [Santana et al. 2016] could be a much lower-cost alternative for experimentation.
- **City Models:** Some authors also argue that it is hard to understand a city and describe an effective and efficient model for it. For example, Wu et al. [Wu et al. 2014] claim that it is necessary to create a useful model of the city to make intelligent decisions. Naphade et al. [Naphade et al. 2011] state that modeling is required to observe and understand the city activity and to avoid generating unnecessary and empty models. Muñoz et al. [Hernandez-Muñoz et al. 2011] state that a unified model of the city is required, so that the huge amount of heterogeneous data generated can be shared among applications and services.
- **Platform Maintenance:** Three works state that deploying and maintaining the platform is a challenge. Perera et al. [Perera et al. 2014] discuss the difficulty of maintaining a middleware to manage millions or billions of devices connected to the platform. Similarly, Wenge et al. [Wenge et al. 2014] discuss that the administration of the platform can be a challenge, due to its size as well as the very large number of devices spread across the city. Hancke et al. [Hancke et al. 2012] point out that

addressing coordination issues in the sensor nodes can be a problem, again because of the size of the city sensor network.

5. REFERENCE ARCHITECTURE FOR SMART CITY PLATFORMS

Based on the knowledge surveyed in this paper, we present a novel, comprehensive reference architecture to guide the development of next-generation software platforms for Smart Cities. The platform was derived from architectures proposed in previous works, with enhancements based on the requirements and challenges described in this survey. First, we describe and analyze the architecture of two platforms presented in the literature: CiDAP and OpenIoT. Then, based on these early works, and on the answers to the research questions presented before, we derived a novel reference architecture. Finally, we compare our proposal with the other two architectures.

5.1. CiDAP

The City Data and Analytics Platform (CiDAP) is a Big Data based platform that aims to use the data collected from the city to enable context-awareness and intelligence in applications and services. This platform processes large datasets collected from an IoT Middleware. Figure 6 presents the architecture of the platform, which has the following five main components.

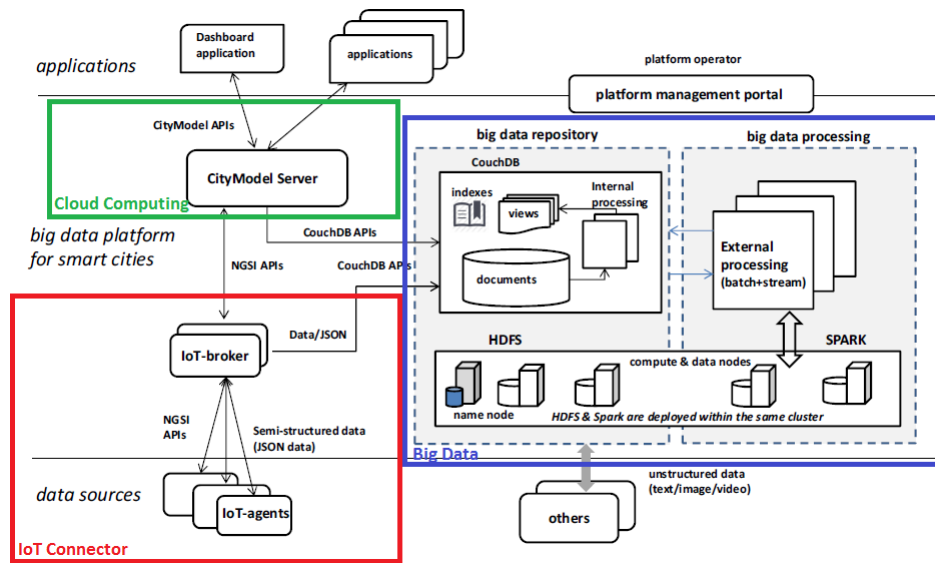


Fig. 6. CiDAP Platform [Cheng et al. 2015]

- **IoT-Agents** connect to the IoT middleware and serve as a gateway to the devices available to the platform. Each data source of the IoT middleware is mapped to an IoT-Agent.
- **IoT-Brokers** act as a unified interface to the IoT agents, facilitating access to the middleware data. This component communicates with the Big Data Repository to send data to be stored, and with the CityModel Server to send data to be used directly by applications.

- The **Big Data Repository** stores raw data collected from the city and processed data from the Big Data processing component. The platform uses the CouchDB⁴ NoSQL database, which stores data as JSON documents. This component also has an internal processing tool that makes processing simple, such as transforming data into new formats, or creating new structured views and tables to index data.
- **Big Data Processing** is responsible for complex or intensive processing using the data stored in the Big Data Repository, such as data aggregation or data mining. Also, it processes historical data using batch processes, or real-time data using data streams. This component uses Apache Spark for this processing.
- **City Model Server** is the interface of the platform to external applications. The CityModel API allows applications to perform simple queries, complex queries, and subscribe to specific pieces of data from the platform. Simple queries request the latest data from devices, complex queries request aggregated historical data, and subscription is a mechanism for applications to periodically receive data from the devices.

The red, green, and blue boxes in Figure 6 highlight the concepts used to implement each layer of the platform. The IoT Connector box has components to facilitate access for IoT devices in the platform. The Big Data box has components to store and analyze the data gathered from multiple sources. Finally, the Cloud Computing box indicates the interface of the platform with external applications, which is implemented using cloud services.

CiDAP is mainly concerned with storing and processing a large amount of data in the platform, which is important because of the massive amount of data collected in a city. The strong points of its architecture are data storage and processing, the real-time and batch processing modules, and the fact that the associated platform was already tested in the SmartSantander testbed.

An important limitation of CiDAP is that the platform does not foresee specific services and tools for application developers, and does not allow the deployment of new services in the platform, making its extensibility difficult.

The red, green, and blue boxes in the figure highlight the concepts used to implement each layer of the platform. The IoT Connector box has components to facilitate the access of IoT devices in the platform. The Big Data box has components to store and analyze the data gathered from multiple sources. Finally, the Cloud Computing box indicates the interface of the platform with external applications which is implemented using cloud services.

CiDAP is mainly concerned with storing and processing a large amount of data in the platform. It is important because of the huge amount of data collected in a city. The strong points of its architecture are the data storage and processing, the real-time and batch processing modules, and the fact that the associated platform was already tested in the SmartSantander testbed.

An important limitation of CiDAP is that the platform does not foresee specific services and tools for application developers and does not allow the deployment of new services.

5.2. OpenIoT

OpenIoT is an Internet of Things platform used by the Vital project [Petrolo et al. 2014] to create a Smart City platform. Figure 7 presents an overview of the platform architecture, which has three layers: the Physical Plane, the Virtualized Plane, and the Utility-App Plane.

⁴<http://couchdb.apache.org>

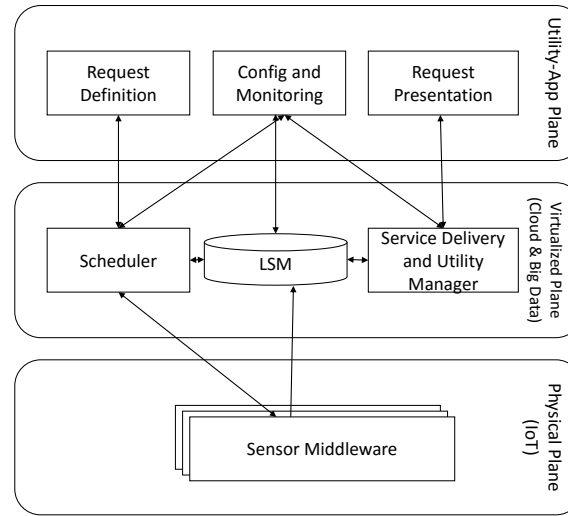


Fig. 7. OpenIoT Platform [Petrolo et al. 2014]

The Physical plane is a middleware responsible for collecting, filtering, combining, and cleaning data from sensors, actuators, and devices. This plane acts as an interface between the physical world and the OpenIoT platform. The current version of OpenIoT uses the X-GSN middleware [Calbimonte et al. 2014], an open-source middleware for managing, monitoring, and controlling IoT devices.

The Virtualized plane aims to store data, execute services, and schedule the execution of these services. The main components of the Virtualized plane are the following:

- The **Scheduler** receives requests for services and ensures the access to resources that the service needs, such as data and data streams. This component is responsible for discovering the sensors required for a service execution.
- The **Cloud Data Storage** keeps all the data from the platform, e.g., data streams collected from the sensors and the data created within the platform, such as user profiles, service definitions, and registered applications. For storing data collected from the IoT middleware, OpenIoT uses the Linked Sensor Middleware (LSM) [Le-Phuoc et al. 2012].
- The **Service Delivery and Utility Manager** has three primary functions: handling the combination of the data collected from the IoT middleware, allowing service definitions, and delivering the results of requested services to the platform or to third-party applications. Also, this component keeps track of the usage of the services defined in the platform for accounting and billing.

The Utility-App Plane, the user interface of the platform, has three main components:

- **Request Definition** enables users to define new applications using the services deployed on the platform, including the definition of service mash-ups.
- **Request Presentation** executes the applications created in the Request Definition component. When a user executes an application, it communicates with the Service Delivery and Utility Manager to retrieve the results from the service executions.

- **Configuration and Monitoring** allows configuration of platform parameters, such as periodicity of sensor data reads and monitoring the health of all platform devices and components.

OpenIoT is a complete platform, handling almost all the main requirements that we described in the survey. The strong points of this platform are the use of an IoT middleware to configure and collect data from devices, the middleware to store the data collected from sensors, the development tools, and the fact that the platform is open source. However, its architecture does not consider other data sources, such as social networks, and does not provide support for pre-processing services relevant when dealing with Big Data.

5.3. The Unified Reference Architecture

Based on the answers to the research questions of this survey, the 23 platforms analyzed, and on the two architectures presented above, we derived a novel reference architecture for Software Platforms for Smart Cities. With this reference architecture, we answer the general research question stated in Section 1 (“What are the elements required for the development of a highly-effective software platform for enabling the easy construction of highly-scalable, integrated Smart City applications?”). Figure 8 presents an overview of the architecture.

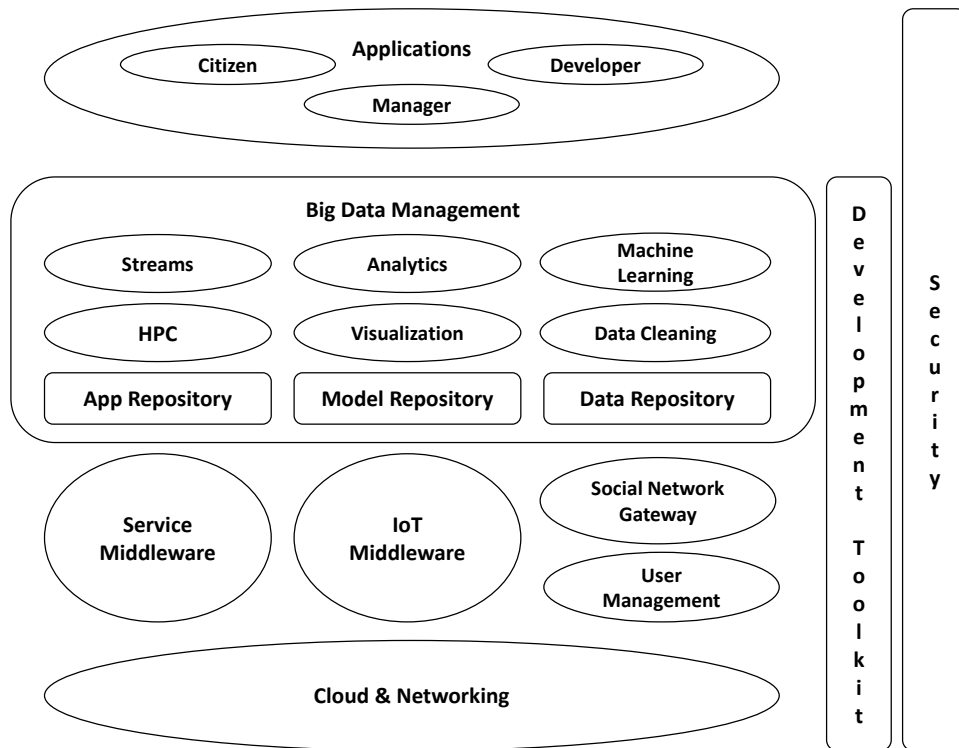


Fig. 8. Reference Architecture for Smart City Platforms

The lowest level component of the reference architecture is **Cloud and Networking**, which is responsible for the management and communication of the city network nodes. This component has to identify all the devices connected to the platform, including servers, sensors, actuators, and user devices. Using cloud computing concepts is important to ensure some fundamental non-functional requirements, including scalability and extensibility.

Just on top of the Cloud and Networking infrastructure, the reference architecture includes the **IoT Middleware** and the **Service Middleware**. The former has to manage the city IoT network and enable the effective communication of the platform with the user devices, city sensors, and actuators. The Service Middleware has to manage the services that the platform will provide to the applications, performing operations such as publishing, enacting, monitoring, composing, and choreographing these services.

The X-GSN middleware can be used to implement the IoT Middleware, which is already used in the OpenIoT project. Another option is to use components of the Sentilo platform, which is also open-source, and implement a complete IoT middleware. The CHOREOS framework [Issarny et al. 2011] can be used to implement the Service Middleware; this project aims to enable choreographies of large-scale service-based software systems.

To provide better services to the citizens, it is important for the platform to store some user data and preferences, which is the role of the **User Management** component. But, to ensure user privacy, this data must be properly protected, and permission to store it must be acquired from the user. Moreover, as the city platform will have many applications, it can be helpful to offer a single sign-on mechanism.

Social networks will have a major role in Smart Cities. They can be used to retrieve data from city conditions, and can be an efficient communication channel between the platform and city government with the citizens. Therefore, it is important to allow the integration of the Smart City platform with existing social networks. This is the role of the **Social Network Gateway**. To implement this gateway, many tools can be used, such as Spark Streaming, which reads data streams of Twitter, and Spring Social, which is a Java-based framework to facilitate the connection with social networks such as Twitter, Facebook, and LinkedIn.

Big Data Management is a module to manage all the data in the platform. It is responsible for storing the data collected from the city and generated by the platform. To this extent, the reference platform has three repositories: (1) an **App Repository** to store applications, including its source/binary code, images, and associated documents; (2) a **Model Repository** to store the city models, such as a traffic model, sensor network model, data model, city maps, and an energy distribution model; and (3) a **Data Repository** to store the data collected from sensors, citizens, and applications. Because of the amount of data that a platform for Smart Cities generate, NoSQL databases can be more suitable than relational databases.

Besides the data storage, the Big Data Management module is also responsible for the processing of city data. There are two types of data processing that might be more suitable for different situations: **Stream processing**, to perform real-time analytics and data-flow processing; and **Batch processing**, to analyze large data-sets. Moreover, this module must be capable of performing useful pre-processing tasks, such as data filtering, normalization, and transformation.

The Big Data module also has a **Machine Learning** component, which facilitates understanding of the city by automatically building models of city processes behavior and making predictions of city phenomena. Since a Smart City will produce an enormous amount of data, a **Data Cleaning** component is responsible for garbage col-

lection, deleting unneeded data, and archiving old data on slower, high capacity data stores.

To implement the Big Data Management components, many open-source tools are available. To the repositories, NoSQL Databases, such as CouchDB, MongoDB, and Cassandra, can store the unstructured or semi-structured data, such as sensor reads and social networks posts. Relational Databases, such as MySQL and PostgreSQL, can store structured data, such as user information and the platform configuration.

To implement the processing engines, many tools are also available. To execute batch processing, Apache Hadoop and Apache Spark are widely used by other platforms. Apache Spark also provides a stream data processing tool, likewise Apache Storm. Many tools offer machine learning algorithms to process large data sets such as Weka⁵, Spark MLlib, and Scikit-Learn⁶.

Relying on aforementioned middleware component, application developers and smart city operators will develop and deploy Smart City applications. By using open data and open services provided by a city, common citizens and users may also execute, or even develop, novel applications to run on top of the city's smart infrastructure. The applications will use the services and data from the platform, but also generate and store data on the platform. The platform should provide an SDK to facilitate the development of applications, including tools such as an Integrated Development Environment (IDE), libraries, and frameworks for commonly used programming languages, and a Smart City Simulator for debugging and experimenting with applications before real deployment.

All components of the platform must support several non-functional requirements, such as scalability, security, privacy, and interoperability. Scalability is fundamental because of the huge amount of devices, data, and services in the platform. Privacy and Security are important because the platform collects, stores, and processes sensible data from the city and citizens. Interoperability will allow the integrated operation of different types of services, devices, and applications. Table V presents options to implement the reference architecture using tools that the platforms described in the survey use.

Table V. Technologies used in the platforms implementation

Component	Tools
IoT Middleware	Sentilo and X-GSN
Data Repository	MongoDB, CouchDB, MySQL, IBM DB2, and Redis
Data Processing	Spark and Hadoop
Stream Processing	Storm
Cluster Management	Apache ZooKeeper and Hadoop YARN
Cloud Environment	OpenNebula and Microsoft Azure
Data Access	REST APIs and Jersey
Security	SAML Protocol
Machine Learning	Weka, Spark MLlib, and Scikit Learn

5.4. Comparison of Architectures

In our architecture, we combined aspects of both platforms described in the beginning of this section. Our Big Data module is similar to the one in CiDAP; both foresee batch

⁵Weka - <https://weka.wikispaces.com/>

⁶Scikit-Learn - <http://scikit-learn.org/stable/>

and real-time processing and big data storage components. However, we added the idea of an application repository (to store data and meta-data associated with applications so that we can better manage and reflect on the applications executing in the city), as well as a model repository (to store different types of models associated with various city-related phenomena such as different kinds of maps, data flows, user behaviors, automated processes, and more).

Similar to OpenIoT, we included a Cloud and Networking layer to manage the devices that collect data from the city and execute service and application components. We also included a service middleware to support many service-related operations, such as deployment, management, composition (via orchestrations and choreographies), and enactment; OpenIoT also provides a Service Delivery component with more limited support for some of these operations.

We also included some components that are not in these two architectures, but were considered relevant in our literature review. The first is the Social Network Gateway, which is important because social networks connect citizens, the city administration, and service providers, and generate a lot of useful data for city applications.

Although OpenIoT provides some development tools to create applications using the available services, a Smart City platform will have to provide a complete software development toolkit. This SDK has to be aware of all the components of the platform, and enable the construction of sophisticated mash-ups based on them. For example, it must allow the development of a service using data from the IoT middleware, combined with data from social networks, and thereby generating a data stream that is filtered, processed, distributed to other users that have subscribed to a specific channel and, later, summarized and stored in a long-term persistent storage in order to maintain historical records.

6. DISCUSSION

We now discuss the findings of this research. Section 6.1 relates the four enabling technologies with the functional and non-functional requirements; Section 6.2 discusses open research challenges; Section 6.3 presents the implications of our survey to Smart City stakeholders, such as city managers, citizens, and developers. Finally, Section 6.4 considers the limitations of this work.

6.1. Enabling Technologies and Requirements

This survey presented multiple approaches for the development of Smart City platforms. From this study, four highly significant functional requirements emerged: management of sensor and actuator networks; management of the data collected from the city; provisioning, management, and development of services; and an environment for the development and deployment of Smart City applications. These features can be related to the enabling technologies, mapping them onto the major functional and non-functional requirements of Smart City platforms.

Technologies around the Internet of Things are used for managing the sensor and actuator networks and their challenges, such as heterogeneity, scalability, and adaptation. Big Data and Cloud Computing are used to deal with the massive amount of data generated from multiple data sources in the city, such as WSN, social networks, and user devices. Big Data tools are required for most data-related activities, such as storing, analyzing, and sharing. Cloud Computing provides a scalable and elastic environment to store and process city data.

Figure 9 shows the relation between the implemented functional requirements from platforms and the enabling technologies. For example, it is possible to verify that most of the Big Data platforms handle Data Management and Data Processing. Cloud Computing platforms handle External Data Access and Service Management.

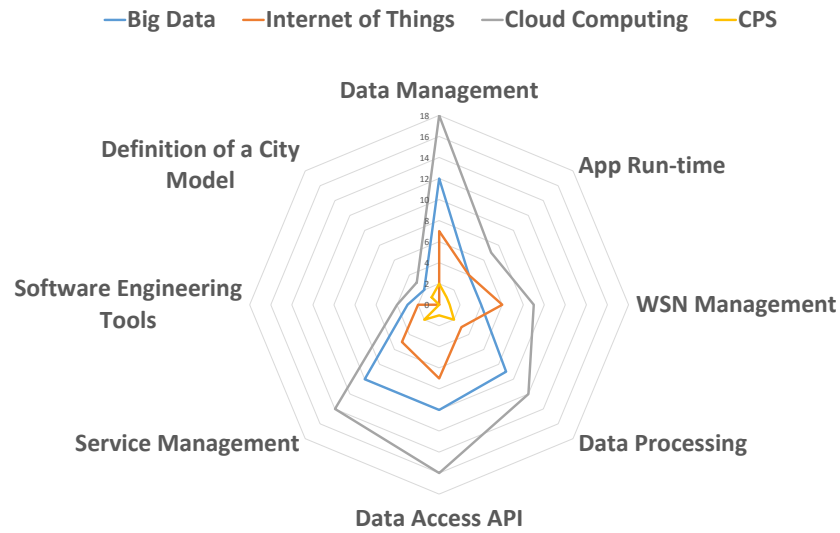


Fig. 9. Relationship between functional requirements and enabling technologies

Figure 10 relates the non-functional requirements and the enabling technologies. We can observe that most platforms are concerned with scalability, regardless of the enabling technology used. It is possible to verify relationships between other non-functional requirements and the technologies. For example, all the CPS platforms handle configurability. Extensibility is mostly offered by platforms that use Big Data, and interoperability mostly by platforms that use IoT.

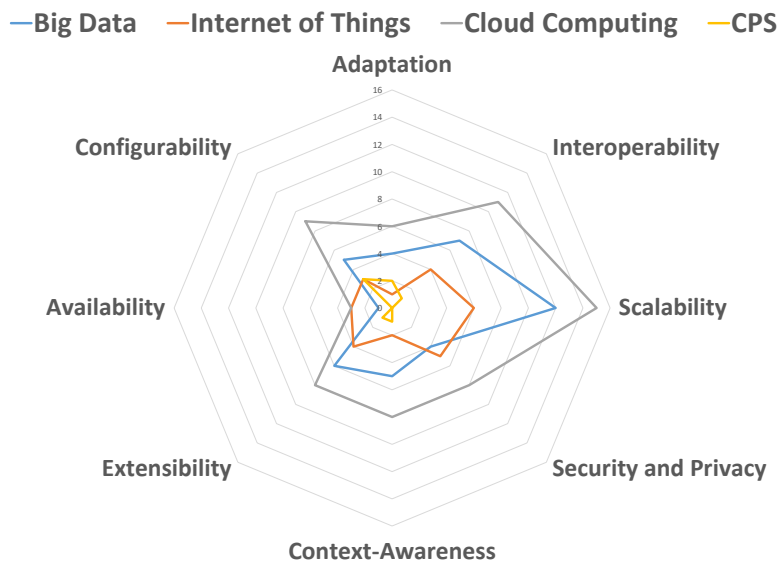


Fig. 10. Relationship between non-functional requirements and enabling technologies

6.2. Challenges and Open Research Problems

Most of the significant challenges and research problems in implementing a platform for Smart Cities is related to data management. The most cited problem in the literature is ensuring the privacy of user data, because of the amount of personal and critical data that a platform needs to handle, such as user locations and medical records.

The second most cited challenge is heterogeneity, because of the large number of different systems, services, applications, and devices that a platform must support. We were surprised that only three authors cited scalability as a problem, because it certainly will be a great challenge support the huge number of devices, users, data, and services in a large metropolis.

An important and understudied issue is how to create a generic platform to support the requirements of different cities. Some literature focuses on a particular city, such as WindyGrid, SmartSantander, and Padova Smart City. Other platforms provide solutions without discussing the characteristics of the cities in which that solution should be applied. The studies proposing generic solutions for Smart Cities lack a discussion concerning how the components of these platforms could be adapted to fit cities of different sizes and characteristics.

6.3. Implications

This paper presented important features that software platforms for Smart Cities should handle. The results give important references for several city stakeholders, such as platform developers, application developers, city managers, system operators, end-users, and Smart City researchers. In this section, we discuss the potential implications of our findings for these stakeholders.

The enabling technologies highlight the infrastructure needed to build Smart Cities. City managers can use this information to improve their investment decisions. Big Data and Cloud Computing deal with an enormous volume of data storage and network infrastructure to access data and services. The city must be equipped with sensors, actuators, and Internet services to take advantage of the Internet of Things and Cyber-Physical Spaces. Besides, the survey can help Smart City application and system developers decide what technologies to use.

The reference architecture highlights the functional and non-functional requirements that platforms and applications developers should consider when developing software for Smart Cities. For platforms developers, this survey indicates that is necessary to deal with big heterogeneous and distributed systems, as well as critical and personal data, in an effective and efficient way. For application developers, the reference architecture shows what kind of services and data they can use to provide better experiences to their end-users. By discussing examples of these systems, we show to end-users, or citizens, the range of system domains that can be developed to facilitate their daily routine, such as urban mobility, air pollution, and health care.

Finally, this survey can also help Smart City researchers by discussing the main open research questions and challenges to be overcome to build smarter cities. These challenges can guide future work in this research area.

6.4. Limitations

In this survey, we decided to describe only the most cited enabling technologies used by Smart City platforms. However, we found other less employed technologies, such as M2M Communications and the Semantic Web. These non-cited technologies are used by few platforms, or are used to solve a small problem but not to serve as a fundamental architectural component of the platform. Thus, there might be key technologies

that end up being very relevant in the future that have not yet been identified in this survey.

We used the most cited paper of each research project to extract components, requirements, and features of the platforms. Other papers, or the project website, may define different aspects.

In this research, we classified the papers according to the enabling technologies only when they were explicitly mentioned. However, we noticed that, in some papers, they were pointed out as a motivating aspect or future work. For example, Khan et al. [Khan et al. 2013] do not explicitly mention IoT in the architecture, but the authors discuss the possibility of using smart hardware such as sensor networks or smart household appliances, which can be organized in an IoT system.

7. RELATED SURVEYS

In our literature search, we found four papers that also surveyed platforms and applications for Smart Cities.

Da Silva et al. [da Silva et al. 2013] surveyed architectures of Smart Cities platforms, analyzing the requirements handled by the platforms. However, they analyzed few platforms and did not distinguish functional and non-functional requirements, and did not address future research and open challenges in the area.

Yin et al. [Yin et al. 2015] conducted a survey on Smart Cities. Although the paper presents some platforms, the main goal of their work was to understand the concept of Smart Cities, identifying the enabling technologies and Smart City research issues.

Al Nuaimi et al. [Al Nuaimi et al. 2015] reviewed the use of Big Data tools and concepts in applications for Smart Cities. The paper mainly presents the relation between the challenges to creating applications for Smart Cities and the use of Big Data tools. It also identifies Smart City requirements that Big Data tools can address. It has some similarities with our work, but we conducted a more general and comprehensive survey.

Finally, Botta et al. [Botta et al. 2015] presented a study of the integration of Cloud Computing and the Internet of Things, defining this novel paradigm as CloudIoT. They describe applications that use this paradigm, such as health care, transportation, and smart cities. The paper presents platforms that use the two concepts, some of which are also presented here, such as OpenIoT and ClouT.

In our work, we studied Smart City software platforms and the related ICT problems, aiming to derive the major functional and non-functional requirements, and the technical and research open challenges. Besides, we presented a reference architecture derived from the requirements pointed out by the surveyed studies.

8. CONCLUSION

Smart City is a concept that has gained increased attention in academic, industrial, and governmental circles. While the urban population is growing, the infrastructure and resources required to support citizens are often insufficient, leading to a degradation in public services. Information and Communication Technologies provide important tools to reduce this problem, helping to improve the sustainable use of resources, city services, and the citizens' quality of life.

Using a software platform rather than *ad hoc* solutions is a more robust and sustainable way to support the features needed by a Smart City environment. In this paper, we surveyed the current research on Smart Cities platforms, aiming to discover their most relevant requirements and how to facilitate the development, integration, and deployment of Smart City applications. We analyzed 23 studies from different groups, proposing multiple approaches for the development of a software platform to answer our general research question "What are the elements required for the development

of a highly-effective software platform for enabling the easy construction of highly-scalable, integrated Smart City applications?”

Based on the analyzed projects, we derived a unified reference architecture supporting the main requirements needed to build a software platform for Smart Cities. Thus, this paper contributes to the state-of-the-art by providing a guide to help software developers and city managers determine the necessary components to handle the functional and non-functional requirements of a software platform for Smart Cities.

The reference architecture is based on the answers of the three research sub-questions. RQ1 (“What are the enabling technologies used in state-of-the-art software platforms for Smart Cities?”) showed us that the Internet of Things, Cloud Computing, Big Data, and Cyber-Physical Systems are the most cited enabling technologies. Answering RQ2 (“What are the requirements that a software platform for Smart Cities should meet?”), we could relate these technologies to the requirements that a software platform should handle. For example, most of the Big Data platforms mention Data Management as a requirement, while Configurability is strongly related to CPS platforms. In this way, an important contribution of this survey is to discuss the requirements that need to be implemented when using a specific enabling technology. In contrast, it helps to decide which technology to use when a specific functional or non-functional requirement is desirable.

Finally, to answer RQ3 (“What are the main challenges and open research problems in the development of next generation, robust software platforms for Smart Cities?”), we presented the most cited challenges and open research problems, according to the literature. These challenges were considered when deriving the reference architecture. In this sense, an important contribution of this survey, especially for developers and researchers of software platforms, is to identify which platform components should be the focus of future work.

This survey described several Smart City initiatives, but all of them are still in their initial phases, posing multiple challenges and open problems that need to be addressed. A collaborative effort of research groups, commercial companies, NGOs, and governments is required to tackle the multitude of scientific, technical, political, and social problems related to the establishment of really-smart cities, reaching the ultimate goal of improving the quality of life of all of a city’s citizens, irrespective of its social and financial situation.

ACKNOWLEDGMENTS

This publication is the result of a project promoted by the Brazilian Informatics Law (Law No. 8248 of 1991 and subsequent updates) and was developed under the Cooperation Agreement 073/2016 between Universidade de São Paulo, FDTE, and Hewlett Packard Enterprise Brazil.

REFERENCES

- Mohammad Aazam, Imran Khan, Aymen Abdullah Alsaffar, and Eui-Nam Huh. 2014. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*. IEEE, 414–419.
- Eiman Al Nuaimi, Hind Al Neyadi, Nader Mohamed, and Jameela Al-Jaroodi. 2015. Applications of big data to smart cities. *Journal of Internet Services and Applications* 6, 1 (2015), 25. DOI: <http://dx.doi.org/10.1186/s13174-015-0041-5>
- Theodoros Anagnostopoulos, Kostas Kolomvatsos, Christos Anagnostopoulos, Arkady Zaslavsky, and Stathes Hadjiefthymiades. 2015. Assessing dynamic models for high priority waste collection in smart cities. *Journal of Systems and Software* 110 (2015), 178 – 192. DOI: <http://dx.doi.org/10.1016/j.jss.2015.08.049>
- Giuseppe Anastasi, Maximiliano Antonelli, Alessio Bechini, Simone Brienza, Eleonora D’Andrea, Domenico De Guglielmo, Pietro Ducange, Beatrice Lazzarini, Francesco Marcelloni, and Armando Segatori. 2013.

- Urban and social sensing for sustainable mobility in smart cities. In *Sustainable Internet and ICT for Sustainability (SustainIT)*, 2013. IEEE, 1–4.
- Wolfgang Apolinarski, Umer Iqbal, and Josiane Xavier Parreira. 2014. The GAMBAS middleware and SDK for smart city applications. In *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2014 IEEE International Conference on. 117–122. DOI: <http://dx.doi.org/10.1109/PerComW.2014.6815176>
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and others. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 50–58.
- Eleana Asimakopoulou and Nik Bessis. 2011. Buildings and Crowds: Forming Smart Cities for More Effective Disaster Management. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2011 Fifth International Conference on. 229–234. DOI: <http://dx.doi.org/10.1109/IMIS.2011.129>
- Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787 – 2805. DOI: <http://dx.doi.org/10.1016/j.comnet.2010.05.010>
- Malcolm Bain. 2014. Sentilo - Sensor and Actuator Platform for smart Cities. (March 2014). Retrieved February 20, 2015 from <https://joinup.ec.europa.eu/community/eupl/document/sentilo-sensor-and-actuator-platform-smart-cities>
- Chitra Balakrishna. 2012. Enabling technologies for smart city services and applications. In *Next Generation Mobile Applications, Services and Technologies (NGMAST)*, 2012 6th International Conference on. IEEE, 223–227.
- Pieter Ballon, Julia Glidden, Pavlos Kranas, Andreas Menychtas, Susie Ruston, and Shenja Van Der Graaf. 2011. Is there a Need for a Cloud Platform for European Smart Cities?. In *eChallenges e-2011 Conference Proceedings, ILMC International Information Management Corporation*.
- Carolina Tripp Barba, Miguel Angel Mateos, Pablo Reganas Soto, Ahmad Mohamad Mezher, and M Aguilar Igartua. 2012. Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights. In *Intelligent Vehicles Symposium (IV)*, 2012 IEEE. IEEE, 902–907.
- Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. 2015. Integration of cloud computing and Internet of Things: A survey. *Future Generation Computer Systems* (2015).
- Nicola Bressan, Leonardo Bazzaco, Nicola Bui, Paolo Casari, Lorenzo Vangelista, and Michele Zorzi. 2010. The Deployment of a Smart Monitoring System Using Wireless Sensor and Actuator Networks. In *Smart Grid Communications (SmartGridComm)*, 2010 First IEEE International Conference on. 49–54. DOI: <http://dx.doi.org/10.1109/SMARTGRID.2010.5622015>
- Nicola Bui and Michele Zorzi. 2011. Health care applications: a solution based on the internet of things. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*. ACM, 131.
- Jean-Paul Calbimonte, Sofiane Sarni, Julien Eberle, and Karl Aberer. 2014. XGSN: An Open-source Semantic Sensing Middleware for the Web of Things. In *7th International Workshop on Semantic Sensor Networks*.
- Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. 2011. Smart cities in Europe. *Journal of urban technology* 18, 2 (2011), 65–82.
- Min Chen, Shiwen Mao, and Yunhao Liu. 2014. Big data: A survey. *Mobile Networks and Applications* 19, 2 (2014), 171–209.
- Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, and Erno Kovacs. 2015. Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander. In *Big Data (BigData Congress)*, 2015 IEEE International Congress on. 592–599. DOI: <http://dx.doi.org/10.1109/BigDataCongress.2015.91>
- Annalisa Cocchia. 2014. Smart and Digital City: A Systematic Literature Review. In *Smart City*, Renata Paola Dameri and Camille Rosenthal-Sabroux (Eds.). Springer International Publishing, 13–43. DOI: http://dx.doi.org/10.1007/978-3-319-06160-3_2
- Louise Coetzee and Johan Eksteen. 2011. The Internet of Things – promise for the future? An introduction. In *IST-Africa Conference Proceedings, 2011*. 1–9.
- Wellington M. da Silva, Alexandre Alvaro, Gustavo H. R. P. Tomas, Ricardo A. Afonso, Kelvin L. Dias, and Vinicius C. Garcia. 2013. Smart cities software architectures: a survey. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 1722–1727.
- Yuri Demchenko, Cees de Laat, and Peter Membrey. 2014. Defining architecture components of the Big Data Ecosystem. In *Collaboration Technologies and Systems (CTS)*, 2014 International Conference on. IEEE, 104–112.
- Salvatore Distefano, Giovanni Merlino, and Antonio Puliafito. 2012. Enabling the cloud of things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012 Sixth International Conference on. IEEE, 858–863.

- Soufiene Djahel, Ronan Doolan, Gabriel-Miro Muntean, and John Murphy. 2014. A Communications-oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches. *Communications Surveys Tutorials, IEEE* PP, 99 (2014), 1–1. DOI: <http://dx.doi.org/10.1109/COMST.2014.2339817>
- A. Elmangoush, H. Coskun, S. Wahle, and T. Magedanz. 2013. Design aspects for a reference M2M communication platform for Smart Cities. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*. 204–209. DOI: <http://dx.doi.org/10.1109/Innovations.2013.6544419>
- Stefan Foell, Gerd Kortuem, Reza Rawassizadeh, Marcus Handte, Umer Iqbal, and Pedro Marrón. 2014. Micro-navigation for Urban Bus Passengers: Using the Internet of Things to Improve the Public Transport Experience. In *Proceedings of the First International Conference on IoT in Urban Space (URB-IOT '14)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 1–6. DOI: <http://dx.doi.org/10.4108/icst.urb-iot.2014.257373>
- MZ Fortes, VH Ferreira, GG Sotelo, AS Cabral, WF Correia, and OLC Pacheco. 2014. Deployment of smart metering in the Búzios City. In *Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA), 2014 IEEE PES*. IEEE, 1–6.
- Jose Antonio Galache, Takuro Yonezawa, Levent Gurgun, Daniele Pavia, Marco Grella, and Hiroyuki Maeomichi. 2014. ClouT: Leveraging Cloud Computing Techniques for Improving Management of Massive IoT Data. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*. 324–327. DOI: <http://dx.doi.org/10.1109/SOCA.2014.47>
- Rudolf Giffinger, Christian Fertner, Hans Kramar, Robert Kalasek, Nataša Pichler-Milanovic, and Evert Meijers. 2007. *Smart cities-Ranking of European medium-sized cities*. Technical Report. Vienna University of Technology.
- Sylva Girtelschmid, Matthias Steinbauer, Vikash Kumar, Anna Fensel, and Gabriele Kotsis. 2013. Big Data in Large Scale Intelligent Smart City Installations. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13)*. ACM, New York, NY, USA, Article 428, 5 pages. DOI: <http://dx.doi.org/10.1145/2539150.2539224>
- Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1645–1660.
- Levent Gurgun, Ozan Gunalp, Yazid Benazzouz, and Mathieu Gallissot. 2013. Self-aware cyber-physical systems and applications in smart buildings and cities. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*. 1149–1154. DOI: <http://dx.doi.org/10.7873/DATE.2013.240>
- Robert E. Hall, B Bowerman, J Braverman, J Taylor, H Todosow, and U Von Wimmersperg. 2000. The vision of a smart city. In *2nd International Life Extension Technology Workshop, Paris*.
- Gerhard P. Hancke, Gerhard P. Hancke Jr., and others. 2012. The role of advanced sensing in smart cities. *Sensors* 13, 1 (2012), 393–425.
- Marcus Handte, Muhammad Umer Iqbal, Stephan Wagner, Wolfgang Apolinarski, Pedro José Marrón, Eva Maria Muñoz Navarro, Santiago Martínez, Sara Izquierdo Barthelemy, and Mario González Fernández. 2014. Crowd Density Estimation for Public Transport Vehicles.. In *EDBT/ICDT Workshops*. 315–322.
- Mohammad Mehedi Hassan, Hanouf Saad Albakr, and Hmood Al-Dossari. 2014. A Cloud-Assisted Internet of Things Framework for Pervasive Healthcare in Smart City Environment. In *Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities (EMASC '14)*. ACM, New York, NY, USA, 9–13. DOI: <http://dx.doi.org/10.1145/2661704.2661707>
- Jos M. Hernandez-Muñoz, Jess Bernat Vercher, Luis Muñoz, Jos A. Galache, Mirko Presser, Luis A. Hernandez Gmez, and Jan Pettersson. 2011. Smart Cities at the Forefront of the Future Internet. Vol. 6656. 447–462. DOI: http://dx.doi.org/10.1007/978-3-642-20898-0_32
- Robert G. Hollands. 2008. Will the real smart city please stand up? Intelligent, progressive or entrepreneurial? *City* 12, 3 (2008), 303–320.
- Aamir Hussain, Rao Wenbi, Aristides Lopes da Silva, Muhammad Nadher, and Muhammad Mudhish. 2015. Health and emergency-care platform for the elderly and disabled people in the Smart City. *Journal of Systems and Software* 110 (2015), 253 – 263. DOI: <http://dx.doi.org/10.1016/j.jss.2015.08.041>
- Valrie Issarny, Nikolaos Georgantas, Sara Hachem, Apostolos Zarras, Panos Vassiliadis, Marco Autili, Marco Aurlio Gerosa, and AmiraBen Hamida. 2011. Service-oriented middleware for the Future Internet: state of the art and research directions. *Journal of Internet Services and Applications* 2, 1 (2011), 23–45. DOI: <http://dx.doi.org/10.1007/s13174-011-0021-3>
- Isam Janajreh, Liu Su, and Fathi Alan. 2013. Wind energy assessment: Masdar City case study. *Renewable energy* 52 (2013), 8–15.

- Rosabeth Moss Kanter and Stanley S. Litow. 2009. Informed and interconnected: A manifesto for smarter cities. (March 2009). Retrieved February 20, 2015 from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1420236
- Zaheer Khan, Ashiq Anjum, and Saad Liaquat Kiani. 2013. Cloud Based Big Data Analytics for Smart Future Cities. In *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*. 381–386. DOI: <http://dx.doi.org/10.1109/UCC.2013.77>
- Zaheer Khan, Ashiq Anjum, Kamran Soomro, and Muhammad Atif Tahir. 2015. Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing* 4, 1 (2015), 1–11. DOI: <http://dx.doi.org/10.1186/s13677-015-0026-8>
- Jaeho Kim and Jang-Won Lee. 2014. OpenIoT: An open service framework for the Internet of Things. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 89–93.
- Danh Le-Phuoc, Hoan Quoc Nguyen-Mau, Josiane Xavier Parreira, and Manfred Hauswirth. 2012. A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web* 16 (2012), 42–51.
- Yong Woo Lee and Seungwoo Rho. 2010. U-city portal for smart ubiquitous middleware. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, Vol. 1. 609–613.
- Pu Liu and Zhenghong Peng. 2013. Smart cities in China. *IEEE Computer Society* 16 (2013).
- Catriona Manville, Gavin Cochrane, Jonathan Cave, Jeremy Millard, Jeremy Kevin Pederson, Rasmus Kåre Thaarup, Andrea Liebe, Matthias Wissner, Roel Massink, and Bas Kotterink. 2014. Mapping smart cities in the EU. (2014).
- Viktor Mayer-Schönberger and Kenneth Cukier. 2013. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.
- Nathalie Mitton, Symeon Papavassiliou, Antonio Puliafito, and KishorS Trivedi. 2012. Combining Cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking* 2012, 1 (2012). DOI: <http://dx.doi.org/10.1186/1687-1499-2012-247>
- G. Mylonas, E. Theodoridis, and L. Munoz. 2015. Integrating Smartphones into the SmartSantander Infrastructure. *Internet Computing, IEEE* 19, 2 (Mar 2015), 48–56. DOI: <http://dx.doi.org/10.1109/MIC.2015.25>
- Milind Naphade, Guruduth Banavar, Colin Harrison, Jurij Paraszczak, and Robert Morris. 2011. Smarter Cities and Their Innovation Challenges. *Computer* 44, 6 (June 2011), 32–39. DOI: <http://dx.doi.org/10.1109/MC.2011.187>
- Rocco Papa, Carmela Gargiulo, and Adriana Galderisi. 2013. Towards an urban planners perspective on smart city. *TeMA Journal of Land Use, Mobility and Environment* 6, 01 (2013), 5–17.
- Anbu Parkavi and Nagarajan Vetrivelan. 2013. A smart citizen information system using Hadoop: A case study. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*. 1–3. DOI: <http://dx.doi.org/10.1109/ICCIC.2013.6724276>
- Charith Perera, Arkady B. Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Sensing as a service model for smart cities supported by Internet of Things. *Trans. Emerging Telecommunications Technologies* 25, 1 (2014), 81–93. DOI: <http://dx.doi.org/10.1002/ett.2704>
- Daniel Pérez-González and Raimundo Díaz-Díaz. 2015. Public Services Provided with ICT in the Smart City Environment: The Case of Spanish Cities. *Journal of Universal Computer Science* 21, 2 (2015), 248–267.
- Riccardo Petrolo, Valeria Loscri, and Nathalie Mitton. 2014. Towards a Cloud of Things Smart City. *IEEE COMSOC MMTTC E-Letter* 9, 5 (Sept. 2014), 44–48. <https://hal.inria.fr/hal-01080273>
- Giuseppe Piro, Iliaria Cianci, Luigi A. Grieco, Gennaro Boggia, and Pietro Camarda. 2014. Information centric services in Smart Cities. *Journal of Systems and Software* 88, 0 (2014), 169 – 188. DOI: <http://dx.doi.org/10.1016/j.jss.2013.10.029>
- Gilles Privat, Mengxuan Zhao, and Laurent Lemke. 2014. Towards a Shared Software Infrastructure for Smart Homes, Smart Buildings and Smart Cities. In *International Workshop on Emerging Trends in the Engineering of Cyber-Physical Systems, Berlin*.
- Junping Qiu, Yanhui Song, and Siluo Yang. 2010. Digital Integrated Model of Government Resources under E-Government Environment. In *Internet Technology and Applications, 2010 International Conference on*. 1–4. DOI: <http://dx.doi.org/10.1109/ITAPP.2010.5566315>
- Aviva Rutkin. 2014. How data can save a city. *New Scientist* 224, 2990 (2014), 24 – 25. DOI: [http://dx.doi.org/10.1016/S0262-4079\(14\)61945-X](http://dx.doi.org/10.1016/S0262-4079(14)61945-X)
- Luis Sanchez, Luis Muoz, Jose Antonio Galache, Pablo Sotres, Juan R. Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, and Dennis Pfisterer. 2014. Smart-Santander: IoT experimentation over a smart city testbed. *Computer Networks* 61 (2014), 217 – 238. DOI: <http://dx.doi.org/10.1016/j.bjp.2013.12.020> Special issue on Future Internet Testbeds Part I.

- Eduardo Felipe Zambom Santana, Daniel Macêdo Bastista, Fabio Kon, and Dejan S Milojicic. 2016. SCSimulator: An Open Source, Scalable Smart City Simulator. In *Tools Session of the 34th Brazilian Symposium on Computer Networks (SBRC)*. Salvador, Brazil.
- A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, E. Marn-Tordera, J. Cirera, G. Grau, and F. Casaus. 2016. Estimating Smart City sensors data generation. In *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 1–8. DOI: <http://dx.doi.org/10.1109/MedHocNet.2016.7528424>
- Kehua Su, Jie Li, and Hongbo Fu. 2011. Smart city and the applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on.* 1028–1031. DOI: <http://dx.doi.org/10.1109/ICECC.2011.6066743>
- Kohei Takahashi, Shintaro Yamamoto, Akihiro Okushi, Shinsuke Matsumoto, and Masahide Nakamura. 2012. Design and implementation of service API for large-scale house log in smart city cloud. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on.* 815–820. DOI: <http://dx.doi.org/10.1109/CloudCom.2012.6427590>
- Kenji Tei and Levent Gurgen. 2014. ClouT: Cloud of things for empowering the citizen clout in smart cities. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on.* IEEE, 369–370.
- Sean Thornton. 2013. Chicagos WindyGrid: Taking Situational Awareness to a New Level. (March 2013). Retrieved February 20, 2015 from <http://datasmart.ash.harvard.edu/news/article/chicagos-windygrid-taking-situational-awareness-to-a-new-level-259>
- United Nations. 2009. Urban and Rural Areas 2009. (2009). <http://www.un.org/en/development/desa/population/publications/urbanization/urban-rural.shtml>
- Athena Vakali, Leonidas Anthopoulos, and Srdjan Krco. 2014. Smart Cities Data Streams Integration: Experimenting with Internet of Things and Social Data Flows. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS'14) (WIMS '14)*. ACM, New York, NY, USA, Article 60, 5 pages. DOI: <http://dx.doi.org/10.1145/2611040.2611094>
- Ignasi Vilajosana, Jordi Llosa, Borja Martinez, Marc Domingo-Prieto, Albert Angles, and Xavier Vilajosana. 2013. Bootstrapping smart cities through a self-sustainable model based on big data flows. *Communications Magazine, IEEE* 51, 6 (2013), 128–134.
- Flix J. Villanueva, Maria J. Santofimia, David Villa, Jess Barba, and Juan Carlos Lopez. 2013. Civitas: The Smart City Middleware, from Sensors to Big Data. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on.* 445–450. DOI: <http://dx.doi.org/10.1109/IMIS.2013.80>
- Eleni I Vlahogianni, Kostantinos Kepaptsoglou, Vassileios Tsetsos, and Matthew G Karlaftis. 2014. Exploiting new sensor technologies for real-time parking prediction in urban areas. In *Transportation Research Board 93rd Annual Meeting Compendium of Papers.* 14–1673.
- Jiafu Wan, Di Li, Caifeng Zou, and Keliang Zhou. 2012. M2M Communications for Smart City: An Event-Based Architecture. In *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on.* 895–900. DOI: <http://dx.doi.org/10.1109/CIT.2012.188>
- Kaiyu Wan, Danny Hughes, Ka Lok Man, and Tomas Krilavicius. 2010. Composition challenges and approaches for cyber physical systems. In *Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on.* 1–7. DOI: <http://dx.doi.org/10.1109/NESEA.2010.5678065>
- Doug Washburn, Usman Sindhu, Stephanie Balaouras, Rachel A. Dines, Nicholas M. Hayes, and Lauren E. Nelson. 2009. Helping CIOs understand smart city initiatives. *Growth* 17 (2009).
- Rong Wenge, Xiong Zhang, Dave Cooper, Li Chao, and Sheng Hao. 2014. Smart city architecture: A technology guide for implementation and design challenges. *Communications, China* 11, 3 (March 2014), 56–69. DOI: <http://dx.doi.org/10.1109/CC.2014.6825259>
- Jules White, Siobhan Clarke, Christin Groba, Brian Dougherty, Chris Thompson, and Douglas C. Schmidt. 2010. R&D challenges and solutions for mobile cyber-physical applications and supporting Internet services. *Journal of Internet Services and Applications* 1, 1 (2010), 45–56. DOI: <http://dx.doi.org/10.1007/s13174-010-0004-9>
- Chao Wu, David Birch, Dilshan Silva, Chun-Hsiang Lee, Orestis Tsinialis, and Yike Guo. 2014. Concinnity: A Generic Platform for Big Sensor Data Applications. *Cloud Computing, IEEE* 1, 2 (July 2014), 42–50. DOI: <http://dx.doi.org/10.1109/MCC.2014.33>
- Shintaro Yamamoto, Shinsuke Matsumoto, Sachio Saiki, and Masahide Nakamura. 2014. Using Materialized View as a Service of Scallop4SC for Smart City Application Services. In *Soft Computing in Big Data Processing*. Springer, 51–60.
- ChuanTao Yin, Zhang Xiong, Hui Chen, JingYuan Wang, Daven Cooper, and Bertrand David. 2015. A literature survey on smart cities. *Science China Information Sciences* (2015), 1–18. DOI: <http://dx.doi.org/10.1007/s11432-015-5397-4>

Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, Vol. 10. 10.

Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of Things for Smart Cities. *Internet of Things Journal, IEEE* 1, 1 (Feb 2014), 22–32. DOI: <http://dx.doi.org/10.1109/JIOT.2014.2306328>