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

EDUARDO FELIPE ZAMBOM SANTANA, University of São Paulo
ANA PAULA CHAVES, Northern Arizona University and Federal Technological University of Paraná
MARCO AURELIO GEROSA, Northern Arizona University and University of São Paulo
FABIO KON, University of São Paulo
DEJAN S. MILOJICIC, Hewlett Packard Labs Palo Alto

Information and communication technologies (ICT) can be instrumental in progressing towards smarter city environments, which improve city services, sustainability, and citizens' quality of life. Smart City software platforms can support the development and integration of Smart City applications. However, the ICT community must overcome current technological and scientific challenges before these platforms can be widely adopted. This article surveys the state of the art in software platforms for Smart Cities. We analyzed 23 projects concerning 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 provides important references to help application developers, city managers, system operators, end-users, and Smart City researchers make project, investment, and research decisions.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Software and its engineering → Software architectures**; **Requirements analysis**; *Middleware*;

Additional Key Words and Phrases: Wireless sensor networks, software platforms

ACM Reference format:
Eduardo Felipe Zambom Santana, Ana Paula Chaves, Marco Aurelio Gerosa, Fabio Kon, and Dejan S. Milojicic. 2017. Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture. *ACM Comput. Surv.* 50, 6, Article 78 (November 2017), 37 pages.
https://doi.org/10.1145/3124391

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq, proc. 465446/2014-0, CAPES proc. 88887.136422/2017-00, and FAPESP proc. 2014/50937-1. 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 University of São Paulo, FDTE, and Hewlett Packard Enterprise Brazil.

Authors' addresses: E. F. Z. Santana, M. A. Gerosa, and F. Kon, Department of Computer Science, University of São Paulo, 1010 Rua do Matão, São Paulo-SP, Brazil; emails: {efzambom, kon, gerosa}@ime.usp.br; A. P. Chaves, Academic Department of Computer, Federal Technological University of Paraná, 1233 Via Rosalina Maria dos Santos, Campo Mourão-PR, Brazil; email: anachaves@utfpr.edu.br; D. S. Milojicic, Hewlett Packard Labs, MS 2213, 3000 Hanover Street, Palo Alto, CA 94304; email: dejan.milojicic@hpe.com.

https://doi.org/10.1145/3124391

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 (i.e., providing innovative urban services, ICT-based or not, to improve citizens' quality of life (Anthopoulos and Reddick 2016)) can help to optimize resource and infrastructure utilization toward increased sustainability. One approach involves creatively combining the vast 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, thereby 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 for a wide range of city services, such as transportation (Djahel et al. 2015), 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 et al. 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. Since they do not interoperate, they lead to duplication of work, incompatible solutions, and non-optimized resource use.

Integrating these domains into a complete and consistent solution requires basic services provided by the underlying software infrastructure. A novel, comprehensive software platform could provide such services, including facilities for application development, integration, deployment, and management to ease 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 the following: 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 this article, we evaluate 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 City Platforms. To do so, we examined 23 Smart Cities software platforms, aiming to answer the following **general research question**:

What characteristics should software platforms provide for enabling the construction of scalable integrated Smart City applications?

We investigated three additional, more specific research questions as follows:

- *RQ1:*. "What are the enabling technologies used in state-of-the-art software plat- forms for Smart Cities?"
- *RQ2:*. "What requirements should a software platform for Smart Cities 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 from the literature the most common enabling technologies employed in platforms for Smart Cities. As detailed 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 3.4. 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, which presents components for implementing a Smart Cities software platform 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 Section 6.

The remainder of this article is organized as follows. Section 2 defines 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 our final conclusions.

2 MAIN CONCEPTS

We first offer various definitions of Smart Cities and then discuss the most adopted enabling technologies for developing software platforms for Smart Cities. We conclude this section by presenting other related surveys and by discussing differences with our work.

2.1 Smart Cities

A “Smart City” has been widely and variously defined. Some definitions 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 infrastructures 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. (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 benchmarks exist for ranking the smartest city using these dimensions.[1]

In their definition of Smart Cities, Washburn et al. (2009) and 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 (2009) declare that creating independent software for each city domain is insufficient for creating Smart City environments. They contend that all city sub-systems (such as transport, education, energy, and water) must be holistically linked in a network to achieve full integration. Definition of Smart Cities by Caragliu et al. (2011) highlights the significant benefit of sustainability and management of natural resources.

[1]Smarts Cities in Europe; http://www.smart-cities.eu.

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

There are also efforts from standard organizations such as International Organization for Standardization (ISO) (ISO/IEC 2015) and International Telecommunication Union (ITU) (ITU-T 2014) to define Smart Cities. ISO's definition focuses on city outcomes to deal with challenges such as climate change, rapid population growth, and political and economic instability. ITU explicitly cites the idea of using ICT technologies to improve life quality, efficiency of urban operation and services, and competitiveness. Both organizations emphasize that these ideas must meet present and future generations' needs.

We align with the vision that, to improve city services and quality of life, a city must have an integrated environment that facilitates interoperability between the city's sub-systems. Thus, in our definition, a Smart City is a city in which Information and Communication Technologies provide social, business, and technological support to deal with city challenges and improve citizens' experiences. In a Smart City, public and private services operate in an integrated, affordable, and sustainable way.

ICT can enable this vision, providing an integrated, unified technological infrastructure, for instance, through a well-designed software platform that can deal with large volumes of data, a wide variety of devices and applications, system interoperability, and other problems related to Smart City environments.

Several countries around the world have already implemented Smart City initiatives, 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 US,[2] 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 1 presents a map with cities that have at least one Smart City project. The map shows that most of the projects exist in developed countries, a few in developing countries, and none in underdeveloped countries, where the need for improvements in urban quality of life is most pressing.

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

[2]10 Smartest Cities in USA; http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america.

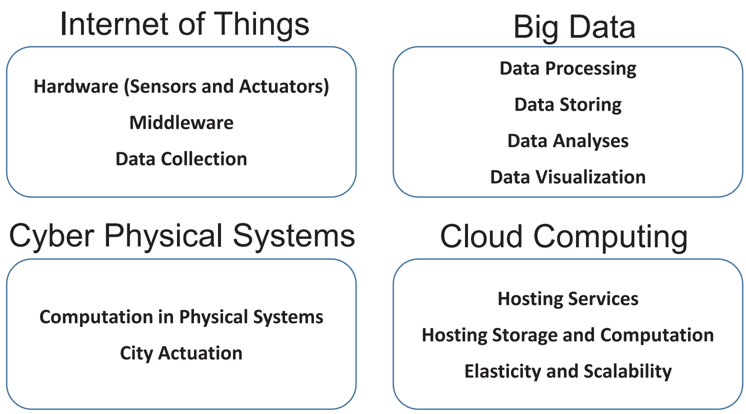

Fig. 2. Platforms for Smart Cities Enabling Technologies.

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 provide an overview and relate them to Smart Cities research. These technologies are used later in this article to group the analyzed platforms and better determine the requirements that the platforms should address.

Lee et al. (2013) describe five requirements (sensing, processing, network, interface, and security) that the technologies used in Smart Cities must handle. IoT and CPS can handle sensing and network requirements, Big Data and Cloud Computing can handle processing, and the interfaces among Smart City services can be handled by services deployed in a Cloud Computing environment. Security is important for all related technologies.

These four technologies are also cited as key enablers to Smart Cities by standard organizations. For instance, ISO and ITU mention Big and Open Data, IoT, and Cloud Computing as important technologies for the implementation of Smart Cities. National Institute of Standards and Technology (NIST)[3] also includes CPS in the related technologies.

Figure 2 presents an overview of the four enabling technologies that we found in our literature survey and examples of how they contribute to Smart City platforms.

Some specific platforms use other useful technologies such as Ubiquitous and Mobile Computing, Machine-To-Machine (M2M) Communications and Service Oriented Architecture (SOA). In this survey, the technologies are used to group the platforms; so, to select the major groups, we only used technologies relevant for more than one platform.

2.2.1 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. (2010) define CPS as the integration of computation with physical processes and promote using 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; applying CPS should introduce such a solution to Smart City applications.

[3] NIST - https://www.nist.gov.

According to Gurgen et al. (2013), CPS enables applications to become aware of the changes in the physical context and adapt their execution according to it.

One Cyber-Physical System related to Smart Cities is WreckWatch (White et al. 2010), an application for detecting traffic accidents. Developed for smartphones, it reads the device's accelerometer and GPS, determining the driver's current speed and acceleration. If WreckWatch detects a strong deceleration, then an accident prediction model analyzes the data and generates an alert to a centralized server if the data indicates an accident.

2.2.2 Internet of Things. IoT describes how objects can become part of the Internet (Coetzee and Eksteen 2011) by becoming typically uniquely identified, with recognized position and status, and accessible to the network. One can define three components in an IoT environment (Gubbi et al. 2013): the hardware, which includes sensors, actuators, and embedded communication circuits; 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 devices. We expect similar components from a Smart City platform.

The very large number of devices used to collect data from cities forces Smart City platforms to use IoT technologies. The data collected from these devices must be transmitted via interconnected networks and then grouped and processed to provide advanced Smart City services. Zanella et al. (2014) present multiple potential uses of the Internet of Things for Smart Cities, for example, 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 energy consumption in Smart Homes.

CPS and IoT are related technologies: CPS deals with the monitoring, coordinating, controlling, and integration of physical entities in information systems (Carruthers 2014), and IoT connects devices (or things) to the Internet infrastructure. Sometimes the two terms are used interchangeably (Salim and Haque 2015); yet, in this survey, we opted to keep them separated, due to the substantial differences between the platforms' goals and requirements addressed by the platforms that refer to these two technologies.

2.2.3 Big and Open Data. Big Data can be considered a set of techniques and tools to store and manipulate large data sets, which conventional technologies, such as relational databases and sequential processing tools, cannot deal with. 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 tools must deal with this challenge. In Smart Cities, the volume of data will be massive and originate from many distributed data sources.
— Variety: Data are collected from different sources and have structured, semi-structured, or unstructured formats, such as video records, relational databases, and raw texts, respectively. This challenge is relevant for Smart Cities, which involve data from cameras, sensors, and citizen's personal 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 be able to rapidly respond to urban problems, such as traffic jams, accidents, and floods.
— Veracity: Because of the large amount of data collected, and the use of multiple data sources, it is important to ensure data quality, since errors in the data or the usage of unreliable sources can compromise the analysis. In cities, poor data sources can include incorrect GPS readings, malfunctioning sensors, and malicious users.

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

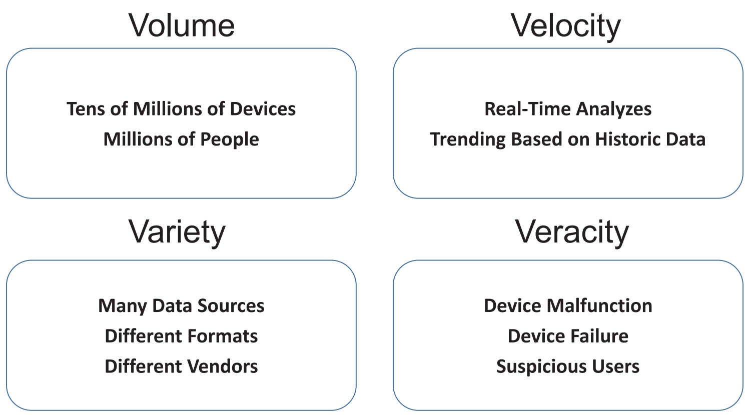

Fig. 3. 4 Vs of Big Data applied to Smart Cities.

Smart Cities already use Big Data tools to support the large 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, such as buses and taxis, continuously send their positions.

Smart City platforms should support Big Data tools, 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 stream processing tools (Girtelschmid et al. 2013), such as Apache Storm; and visualization tools (Khan et al. 2013), such as RapidMiner. Al Nuaimi et al. (2015) discuss potential applications of Big Data tools in Smart Cities, such as recognizing traffic patterns and avoiding traffic jams, facilitating the decisions of city governments using analyses of large datasets, and predicting the use of resources, such as electricity, water, and gas.

Another fundamental aspect of data in Smart Cities is the concept of *Open Data* (Janssen et al. 2012). By granting citizens, companies, and NGOs open access to municipal data, cities can leverage their innovation ecosystems to produce novel solutions to city problems. Many cities around the world have opened large collections of datasets related to health, education, mobility, real estate, and so on (Hielkema and Hongisto 2013; Schaffers et al. 2011). Anyone with creativity and the required technical skills may develop useful applications for citizens, businesses, and/or the city government.

There has been growing discussion about how to make data open, useful, and effective, not only in the context of Open Government (Lathrop and Ruma 2010) and Smart Cities but also in Science in general (Molloy 2011). Effective Open Data must be structured around well-defined standards and accompanied by corresponding meta-data, and it must be made available in files in standard formats or via well-defined APIs. The license associated with the data must clearly state whether it can be shared and freely used by third parties. Finally, proper care must be taken with regard to the privacy of citizens, applying anonymization techniques whenever needed.

Among others, examples of cities that have embraced open data include the following: Dublin (Stephenson et al. 2012), Barcelona,[4] and Chicago[5]; these cities already have significant Open Data

[4]Open Data Barcelona; http://opendata.bcn.cat/opendata/en.
[5]City of Chicago, Data Portal; https://data.cityofchicago.org.

portals to leverage citizens' e-participation and enable startups and other companies to provide innovative solutions.

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, Cloud Computing can support the underlying infrastructure reconfiguration necessary for a Smart City's highly dynamic environment.

Many authors (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 environment, as some Smart City projects currently do (Mitton et al. 2012; Tei and Gurgen 2014).

Another concept related to cloud computing environments in Smart Cities is Soft- ware as a Service (SaaS), which provides sensor data with a cloud computing infrastructure. Perera et al. (2014) extend this concept with the term Sensing as a Service. The ClouT platform (Tei and Gurgen 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, using Big Data tools, the data generated from an IoT middleware can be stored and processed in a cloud environment. This synergistic combination supports important non-functional requirements such as scalability, elasticity, and security.

2.3 Related Surveys

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

da Silva et al. (2013) surveyed architectures of Smart City Platforms, analyzing the requirements handled by the platforms. However, they analyzed only a few platforms, did not distinguish functional and non-functional requirements, and did not address future research and open challenges in the area.

Yin et al. (2015) survey on Smart Cities presents some platforms, yet their main goal was to understand the concept of Smart Cities by identifying the enabling technologies and Smart City research issues.

Al Nuaimi et al. (2015) reviewed the use of Big Data tools and concepts in applications for Smart Cities, mainly presenting the related challenges between the two and identifying Smart City requirements that Big Data tools can address. While there are similarities with our work, we conducted a more general and comprehensive survey.

Kakarontzas et al. (2014) present the results of a questionnaire passed to 18 expert engineers from various Smart City projects, aiming to discover quality properties for Smart Cities to propose a conceptual architecture for a Smart City framework. The engineers answered questions about architectural design, data sources, and management, as well as about managing and funding their solutions. Using the data collected in the survey, the authors derived functional and quality requirements for Smart Cities, which relate to the requirements we present in Section 3.4.2. Finally, based on the derived requirements, they presented a conceptual architectural framework, selecting a set of architectural patterns to handle Smart City requirements.

Botta et al. (2016) present 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 article presents platforms that use the two concepts, some of which are also presented here, such as OpenIoT and ClouT.

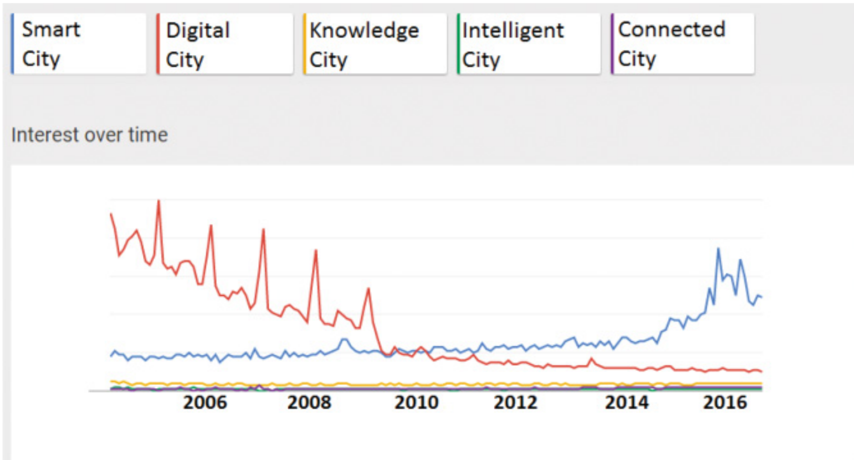

Fig. 4. Trends in Smart City related terms.

Guo et al. (2015) discuss Mobile Crowd Sensing and Computing (MCSC), a paradigm based on the use of data acquired by user-companioned devices such as mobile phones and vehicles. MCSC benefits many urban application domains, such as public safety and environment monitoring. Although few of the platforms we found in our survey directly cite MCSC, it is also relevant in Smart City research. Many of the application domains, challenges, and technologies from our survey are similar to those in Guo et al. (2015).

The cloud platform in a cloud-centric IoT architecture provides storage resources for aggregated sensing data, as well as computing resources for data analytics and data mining for information retrieval and knowledge discovery on sensing data received via IoT objects, namely the built-in sensors of mobile smart devices.

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

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 articles 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.

Since the expression “Digital City” is still used, we analyzed the definition of this expression and its 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 (2014) and Yin et al. (2015).

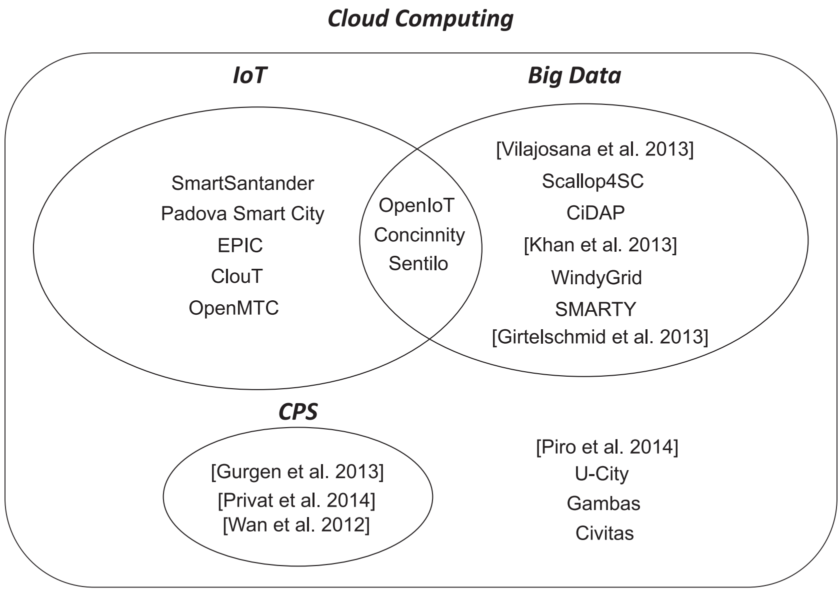

Fig. 5. Use of Enabling Technologies by Smart City Platforms.

The next section describes existing platforms, developed as research projects with different approaches. Section 3.3 shows systems developed using these platforms. Finally, in Section 3.4, we present a set of functional and non-functional requirements extracted from our analysis of the platforms and systems described in the previous sections.

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, most commonly IoT and Big Data.

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 placed 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 300 sensors, the platform collects environmental data, such as CO_2 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 Wireless Sensor Networks (WSN). This middleware aims to deal with the heterogeneity, interoperability, scalability, extensibility, and reconfigurability 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.

Open Machine Type Communications (OpenMTC) (Elmangoush et al. 2013) is a Machine-To-Machine- (M2M) based communication platform for Smart Cities. Its goal is to enable efficient communication among many 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 aforementioned platforms led to identifying four major functional requirements: WSN management, management of the data collected from the city, services and applications management, and an infrastructure to make the data from the platform available to city applications. The analysis also led to identifying five non-functional requirements: adaptation, interoperability, scalability, extensibility, and reconfigurability.

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 of security concerns.

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

OpenIoT[6] 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) is built 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.

Sentilo (Bain 2014) is a platform that deals with the management of sensors and actuators, designed for Smart Cities that desire 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. Originally designed for Barcelona, after its deployment, the City released the code for the Sentilo project 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 lifecycle (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.

[6] OpenIoT; https://github.com/OpenIotOrg/openiot.

A weak point of these platforms is the lack of stream 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. Despite the privacy problems, offering context-aware, customized services to citizens is highly desirable.

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

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 that allows 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.

SCALable LOgging Platform for Smart City (Scallop4SC) (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. It uses the MapReduce algorithm to achieve the objective of analyzing smart building data.

CiDAP (Cheng et al. 2015) is a Big Data analytics platform deployed into the SmartSantander testbed. The platform uses data collected from SmartSantander and analyzes it to understand the city's behavior. 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 that aims 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 retrieve 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. (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 nonfunctional requirements, we identified scalability and adaptation.

Most of the platforms in this section lack 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. Most also lack a discussion about security concerns.

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

Piro et al. (2014) present a two-layered service platform for creating 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 city data.

U-City (Lee and Rho 2010) is a platform for creating smart ubiquitous cities that 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 platform's management.

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 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 City services by facilitating the development and deployment of Smart City applications and to avoid the emergence of "information islands" (Qiu et al. 2010), that is, disconnected applications that do not share relevant information. Citizens connect to the middleware via a special device called the Civitas Plug, which ensures privacy and security. The middleware has two main design principles to facilitate 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 work with more than 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 platform maintenance difficult. 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. (2013) present a middleware for Smart Cities autonomic 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, which provides application developers with the contexts of both individual users and the city.

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. (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 mission-critical wireless messages.

The main functional requirements identified for this group of platforms were autonomic 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 in 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 Commercial Smart City Initiatives

Big ICT companies, such as IBM, Microsoft, and Oracle are working on Smart City initiatives. However, in our literature and web searches, we found very little technical information about R&D efforts by the major players. We located a few documents describing Smart City Platforms designed by Cisco and Oracle, and Microsoft and IBM have white papers describing their vision about Smart Cities and solutions already deployed in cities around the world.

Oracle has a Smart City solution based on three platforms.[7] The *Smart Innovations Platform* enables the communication of the city government with the population via chat, phone, and e-mail. The *Smart Process Platform* for continuous monitoring and improvement of city services, helping to identify which services to prioritize, extend, consolidate, or discontinue. The *Smart Infrastructure Platform* enables the integration and interoperability of legacy IT infrastructure and new city services. Many cities around the world, such as New York, Madrid, and Hong Kong, already use Oracle Smart City solutions, according to Oracle.

Cisco is working with cities, such as Amsterdam and Nice, to deploy an IoT infrastructure.[8] In Nice, the city government implemented, with Cisco's support, a four-layered Smart City platform to collect city data. *Layer 1* comprises sensors and the networking infrastructure to gather and transmit the data. *Layer 2* is responsible for processing, storing, and analyzing data, which occurs in distributed points across the city and thereby boosts the platform's scalability. *Layer 3* is a central computation infrastructure enabling the integration, storage, and sharing of city data. *Layer 4* comprises Smart City applications developed using the platform services and data. Cisco also presents a list of non-functional requirements that are necessary to handle a Smart City platform, such as security, extensibility, scalability, flexibility, and interoperability.

The Hitachi vision of Smart Cities presents three main phases in the ITC infrastructure (Kohno et al. 2011). First, the data are collected from households, buildings, and other end-user devices. Second, the data are analyzed by information systems. Finally, real-time data are provided for Smart City services and applications. Hitachi is already working on many Smart City projects in Japan, mainly in energy and water distribution.

Microsoft's CityNext project[9] presents ideas for developing Smart Cities initiatives, such as the use of Cloud Computing, Big Data, Internet of Things, and Social Networks. They also introduce four main objectives: engaging the city population, empowering city employees to increase productivity and efficiency, transforming the city with new digital services, and optimizing city operations and infrastructure. Some cities that already use Microsoft services are Buenos Aires,

[7] Oracle's City Platform Solution; https://goo.gl/5q5Ufz.
[8] The Internet of Everything for Cities; http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/everything-for-cities.pdf.
[9] Microsoft CityNext; https://enterprise.microsoft.com/en-us/industries/citynext/.

Table 1. Domains of Smart City Systems

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

which developed a city dashboard, Tacoma, with an education analytics and research system, and Helsinki, which developed a solution to collect and analyze data from bus sensors to reduce fuel consumption. We could not find any detailed system or platform software architecture from Microsoft.

IBM has many different Smart Cities projects, among the most cited of which is the Intelligent Operations Center that helps cities to monitor and manage resources, incidents, and events in real time. One important requirement of this system is the integration of systems and data from the various city departments and legacy systems. However, we were unable to find relevant technical information about IBM's solutions.

Some of the academic projects presented in Section 3.1 are also supported by ITC companies, such as SmartSantander, which is supported by Telefonica, Alcatel, and Ericsson, and EPIC, which is supported by IBM.

3.3 Specific Systems

In this section, we illustrate specific systems and applications built on top of the platforms presented in the previous section. The most common systems domains that we found were *City Sensing*, applications to monitor city conditions such as temperature and humidity; *Traffic Control*, including streets surveillance and public transportation monitoring; and *Public Safety* with applications to prevent disasters and crimes. Table 1 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. (2014) present a context-aware urban bus navigator to help travelers find the best buses for their trips. Handte et al. (2014) describe a system that estimates the number of passengers on 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. (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 (2011) present a health care system that monitors patients' conditions, sends their data directly to doctors, and calls emergency services if the patient has an urgent problem. Bressan et al. (2010) present a monitoring application to manage and collect data from all the light posts connected to the platform.

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 CO_2 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 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 logs of emergency (911) calls, traffic conditions, public building information, and surveillance cameras.

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. (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 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 using sensors such as air pollution and temperature. Most of the applications are developed externally to the platform, using only one or more platform services.

3.4 Requirements for Smart City Software Platforms

To answer the second research question *"What requirements should a Smart Cities software platform 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.

3.4.1 Functional Requirements. The main goal of a Smart Cities platform is to facilitate the development of Smart City applications. Towards this aim, most of the analyzed platforms implement requirements for collecting, managing, and sharing city data and for providing tools to facilitate the development of Smart City applications. Table 2 presents an overview of the functional requirements for Smart City platforms, which we describe in the following:

— **Data Management:** Most Smart Cities platforms implement this requirement, which includes collection, storage, analysis, and visualization of city data. The analyzed platforms use different techniques for this requirement, such as relational databases (Hernández-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).

Table 2. Functional Requirements for Smart City Platforms

	Data Management	Application Runtime	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

The most cited functional requirements are related to the data stored in the platform (Data Management, External Access, and Processing) and the management of resources of the platform (WSN, Data, and Services).

- **Applications Runtime:** Some platforms focus on managing the execution of their applications, aiming to facilitate the applications' deployment and integration. Some platforms provide a complete environment for developers to deploy their applications (Apolinarski et al. 2014); while others offer an execution runtime 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 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 (Hernández-Muñoz et al. 2011; Tei and Gurgen 2014). Other platforms (Bain 2014) do not explicitly mention this but indeed include 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 3,000 sensors, and SmartSantander (Sanchez et al. 2014), with more than 20,000 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 datasets, 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) analyze data streams in real time.
— **External Data Access:** Almost all platforms describe an interface for external applications to access the platform data, most commonly an API. Some platforms use REST (Hernández-Muñoz et al. 2011; Elmangoush et al. 2013), while 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, one platform (Gurgen 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 composition 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 city model 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. (2015), the city model is used to allow queries in the data from the city sensor network. Privat et al. (2014) use a finite-state model to represent the possible city data flows.

Based on the aforementioned functional requirements, we can observe that the main platform's activities aim to control the city data lifecycle: (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.

3.4.2 Non-Functional Requirements. Most of the non-functional requirements of Smart City platforms relate to large, heterogeneous distributed systems, such as scalability, adaptation, and interoperability. Other non-functional requirements relate to the manipulation of critical and personal data from citizens, such as security and privacy. Table 3 presents an overview of the non-functional requirements for Smart City platforms, which we describe in the following.

Table 3. 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

The most cited non-functional requirements are interoperability, security, and scalability.

- **Interoperability:** Different devices, systems, applications, and platforms comprise a Smart City environment, all of which must operate in an integrated fashion and may include sensors from multiple vendors, systems implemented in different languages, platforms that share data and users, and legacy systems should communicate with the new platforms. Previous work in the field adopted several techniques to handle this need, including 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:** A Smart City platform's number of users and services and amount of data will be massive and 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); and CiDAP collected more than 50GBs 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; Hernández-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 consumption 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 such as user information (Apolinarski et al. 2014; Privat et al. 2014) like location, activity, and language, or city information (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 or changing the route of a user to avoid polluted areas.
— **Adaptation:** Related to context awareness, many platforms adapt their behavior to context to achieve fault-tolerance, choose a closer server to improve efficiency, decide for batch or real-time processing or adapt data from multiple data sources. Adaptation 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 for assuring that it meets evolving system requirements and user needs. Hernández-Muñoz et al. (2011) state that easy extensibility is valuable, because one cannot know in advance what services a city will need in the future. 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 platform's many variables. Two platforms (Wan et al. 2012; Privat et al. 2014) highlighted the importance of self-configurability capacities given the massive number of configurations a Smart City platform needs. Other platforms (Lee and Rho 2010; Kim and Lee 2014) provide a portal to centralize the configurations.

Based on the above non-functional requirements, 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 for all data requirements; extensibility, which is required for service management; and configurability, which is also important for all functional requirements. The non-functional requirements are similar to the challenges and open research problems we present in the next section.

Table 4. 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, 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.	Dashboards, controllers, and control loops.
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 insufficient 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.

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 4 presents an overview of the main challenges, which we describe in the following.

— **Privacy:** is the most cited challenge for implementing a Smart City platform; the main reason, as Hassan et al. (2014) and Balakrishna (2012) show, is that the data collected from the city include personal, enterprise, and governmental data that unauthorized users should not be able to access. 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 must 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. 2015; Perera et al. 2014). Data Analysis is also a challenge, because it is difficult to extract useful knowledge (Hassan et al. 2014). Another challenge is data trustworthiness; for example, Wu et al. (2014) claim that a high number of data sources makes it difficult to ensure that all data are correct.

— **Heterogeneity:** This is a challenge, because the different devices in a Smart City generate diverse data (Wu et al. 2014; Su et al. 2011; Wan et al. 2012). Naphade et al. (2011) raise the problem of managing data across all city systems because of these variations in data. Other authors (Wenge et al. 2014) state that a Smart City platform should define standards across heterogeneous devices, systems, and domains.

— **Energy Management:** Some authors cite Energy Consumption as a challenge all platform components must face, such as sensors, actuators, and servers (Perera et al. 2014). Moreover, Hassan et al. (2014) point out that energy management in a Smart City health care application is vital, because applications or services in domains like this cannot fail during 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. (2015) explain that good communication mechanisms are required to share platform data with applications.

— **Scalability:** In the coming 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 during special events in the city. Su et al. (2011) discuss how a Smart City platform must support large-scale, efficient services. As an example, Sinaeepourfard et al. (2016) estimated that the city of Barcelona will need more than 1 million sensors to cover the entire city, generating more than 8GB of data every day.

— **Security:** Unauthorized users accessing city services without permission can cause serious harm. Hancke et al. (2012) consider whether city networks will be safe from cyber-terrorism and cyber-vandalism. Gurgen et al. (2013) highlighted the importance of security in CPS platforms, as such systems control aspects of the city infrastructure, which a malicious user can corrupt, for example, by tampering with traffic lights and light posts.

— **Lack of Testbed:** The lack of testbeds is cited by Elmangoush et al. (2013) and Hernández-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 fully understand a city and describe an effective and efficient model for it. For example, Wu et al. (2014) claim that it is necessary to create a useful model of the city to make intelligent decisions. Naphade et al. (2011) state that modeling is required to observe and understand city activity, as well as to avoid generating unnecessary and empty models. Hernández-Muñoz et al. (2011) state that a unified model of the city is required so the massive 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. (2014) discuss the difficulty of maintaining a middleware to manage millions or billions of devices connected to the platform. Similarly, Wenge et al. (2014) discuss that administration of the platform can be a challenge, due to its size and many devices spread across the city. Hancke et al. (2012) point out that addressing coordination issues in the sensor nodes can be a problem, again because of the city sensor network size.

5 REFERENCE ARCHITECTURE FOR SMART CITY PLATFORMS

Based on the knowledge surveyed in this article, 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, which we chose because they are complementary: CiDAP and OpenIoT. The CiDAP focuses on the IoT network and data collection, and OpenIoT focuses on data storage and processing. Based on these early works, and on the

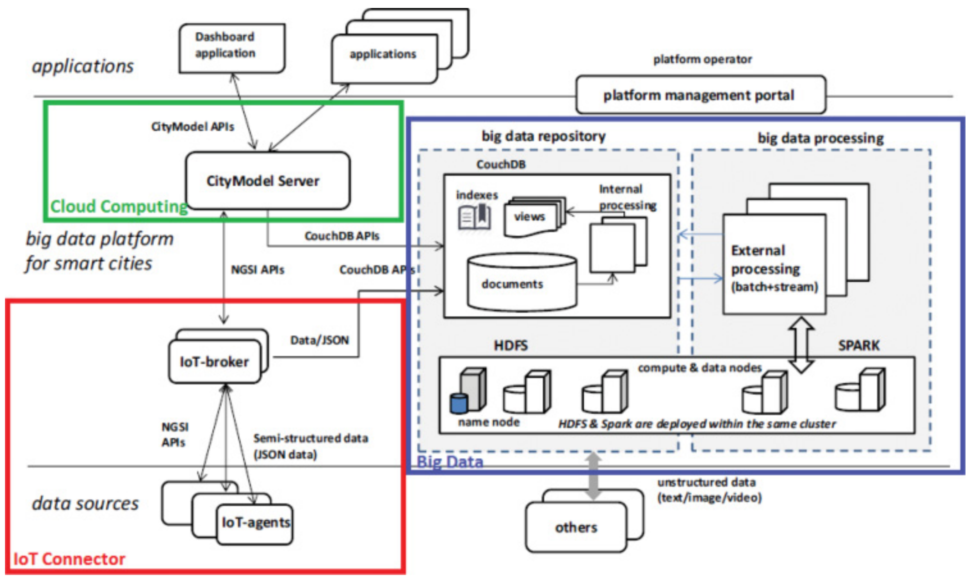

Fig. 6. CiDAP Platform (Cheng et al. 2015).

answers to the research questions presented above, we derived a novel reference architecture. Finally, we compare our proposal to 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 city 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:

- **IoT-Agents** connect to the IoT middleware and serve as a gateway to the devices available to the platform; each data source 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 CouchDB[10] NoSQL database, which stores data as JSON documents. This component includes an internal tool that simplifies processing, such as transforming data into new formats or creating new structured views and tables to index data.
- **Big Data Processing** uses Apache Spark for complex, intensive processing of the data stored in the Big Data Repository, such as data aggregation or data mining. It also processes historical data using batch processes or real-time data using data streams.
- **City Model Server** is the platform's interface to external applications. The CityModel API allows applications to perform simple queries and complex queries and subscribe to specific pieces of data from the platform. Simple queries request the latest data from devices,

[10] http://couchdb.apache.org.

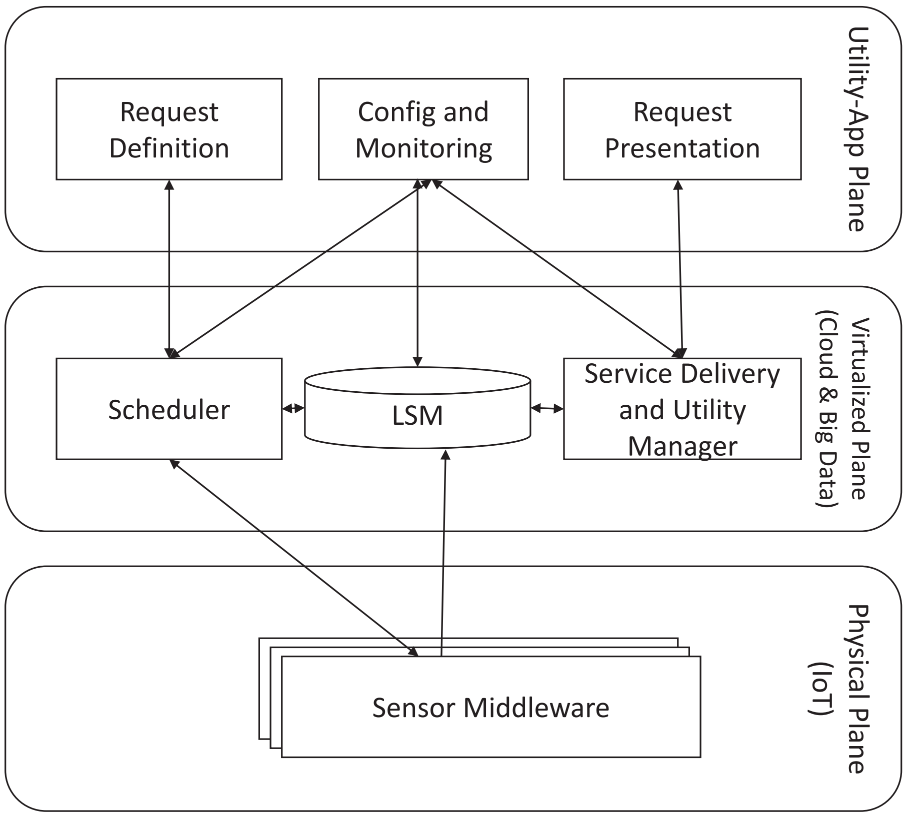

Fig. 7. OpenIoT Platform (Petrolo et al. 2014).

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 primarily stores and processes a large amount of data, 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, 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.

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.

The Physical plane is a middleware 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 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** saves all the data from the platform, for example, 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 combined 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 International Standards

International standardization organizations such as ISO and ITU, and national institutes such as NIST are also working on Smart Cities. This is significant, since one of the main challenges in Smart Cities is the interoperability among multiple infrastructures and systems.

ISO/IEC issued a preliminary report (ISO/IEC 2015) covering technological aspects in Smart Cities. The document discusses the need for a Smart City framework that connects all the city processes; a city model to enable the interoperability and understanding of the city; a data and service model to enable the exchange of data and service reuse throughout the city infrastructure; and data flows to describe all the data phases, such as collection, storage, analysis, and visualization. The report also defines a series of technologies that can be used for the implementation of Smart City platforms and applications, such as Cloud Computing, Internet of Things, Big and Open Data, Embedded Networks, and Service-Oriented Architecture.

ITU proposes a four-layer platform to sense the city using IoT devices,[11] transmit data from the devices to the city ICT infrastructure, analyze and store city data, and provide data and services to city applications. The four layers of the ITU platform are as follows:

— **Sensing Layer**, which consists of terminal nodes and capillary networks. Examples of terminal nodes are sensors, actuators, cameras, and RFID readers. They monitor and control the city physical infrastructure. The capillary network connects the terminal nodes to the Network Layer.
— **Network Layer**, which is the interconnection of the various city networks provided by the city's telecommunication operators.
— **Data and Support Layer**, which stores data collected by the city terminal nodes and data from other city services and applications. This layer is also responsible for providing services and data (raw and analyzed) to the city applications.
— **Application layer**, which includes various services and applications available for citizens, managers, and companies.

NIST is also working on the definition of an IoT-based Smart City Framework with the objective of providing a consensus taxonomy and common architectural principles to support portable and interoperable Smart City services and applications.[12] To define the requirements of the Smart City Framework, NIST first surveyed Smart City applications areas such as Smart Homes, Smart Grids, waste collection, and traffic management. After analyzing these areas, the authors proposed a set of requirements for the Framework, which are very similar to the ones we found in our survey and includes sensing and action capabilities, data management, service management, and security.

5.4 The Unified Reference Architecture

Based on the answers to the survey research questions, the 23 platforms analyzed, the documents of the standard organizations, and the two architectures presented above, we derived a novel reference architecture for Software Platforms for Smart Cities, as shown in Figure 8. With this reference architecture, we aim to identify the elements required for developing a highly effective software platform that can enable the construction of highly scalable, integrated Smart City applications.

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 should identify all the devices connected to the platform, including servers, sensors, actuators, and user devices. Using cloud computing concepts is important for ensuring some fundamental non-functional requirements, including scalability and extensibility.

Directly on top of the Cloud and Networking infrastructure, the reference architecture includes the *IoT Middleware* and the *Service Middleware*. The former manages the city IoT network and enables the platform's effective communication with the user devices, city sensors, and actuators. The Service Middleware manages 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; it aims to choreograph large-scale service-based software systems.

[11] Overview of smart sustainable cities infrastructure; https://goo.gl/GJ2o05.
[12] NIST IES-City Framework Library; https://pages.nist.gov/smartcitiesarchitecture/library/.

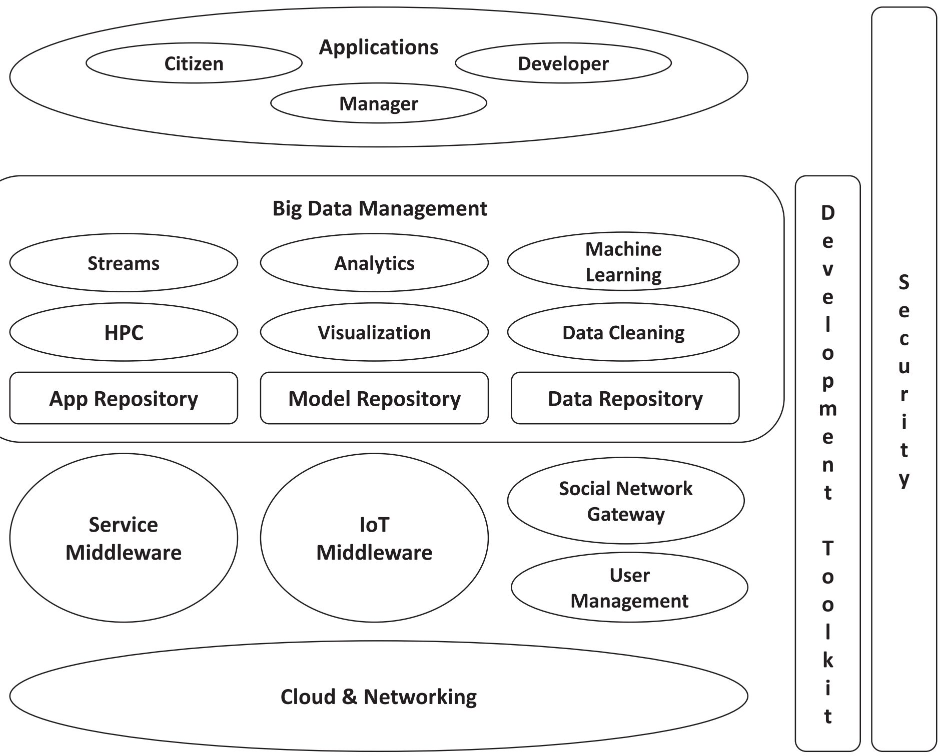

Fig. 8. Reference Architecture for Smart City Platforms.

To provide better services to citizens, it is important for the platform to store some user data and preferences, which is the role of the *User Management* component. However, to ensure user privacy, these 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 play 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 city government and 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 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 generates, NoSQL databases may be more suitable than relational databases.

Besides data storage, the Big Data Management module is also responsible for the processing 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 datasets. Moreover, this module performs 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 process behaviors and predicting city phenomena. Since a Smart City will produce an enormous amount of data, a **Data Cleaning** component is responsible for garbage collection, 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. NoSQL Databases, such as CouchDB, MongoDB, and Cassandra, can store the unstructured or semi-structured data to the repositories, 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 and Apache Storm also provide stream data processing tools. Many tools offer machine-learning algorithms to process large datasets such as Weka,[13] Spark MLib, and Scikit-Learn.[14]

Relying on the aforementioned middleware component, application developers and Smart City operators 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 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 massive amount of devices, data, and services in the platform. Privacy and Security are critical, 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 5 presents options to implement the reference architecture with tools used by the platforms described in the survey.

5.5 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 and real-time processing and Big Data storage components. However, we added the concept of an application repository (to store data and meta-data associated with applications so 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).

Like 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

[13] Weka; https://weka.wikispaces.com/.
[14] Scikit-Learn; http://scikit-learn.org/stable/.

Table 5. 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 Haddop YARN
Cloud Environment	OpenNebula and Microsoft Azure
Data Access	REST APIs and Jersey
Security	SAML Protocol
Machine Learning	Weka, Spark MLib, and Scikit Learn

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 need to provide a complete software development toolkit. This SDK should be aware of all the components of the platform and based on them enable the construction of sophisticated mash-ups. For example, it must allow the development of a service using data from the IoT middleware, combined with data from social networks, and thereby generate a data stream that is filtered, processed, distributed to other users who have subscribed to a specific channel, and, later, summarized and stored in a long-term persistent storage 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; and 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 developing Smart City platforms. From this study, four highly significant functional requirements emerged: management of sensor and actuator networks; management of data collected from the city; provision, management, and development of services; and an environment for development and deployment of Smart City applications. These features can be compared 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

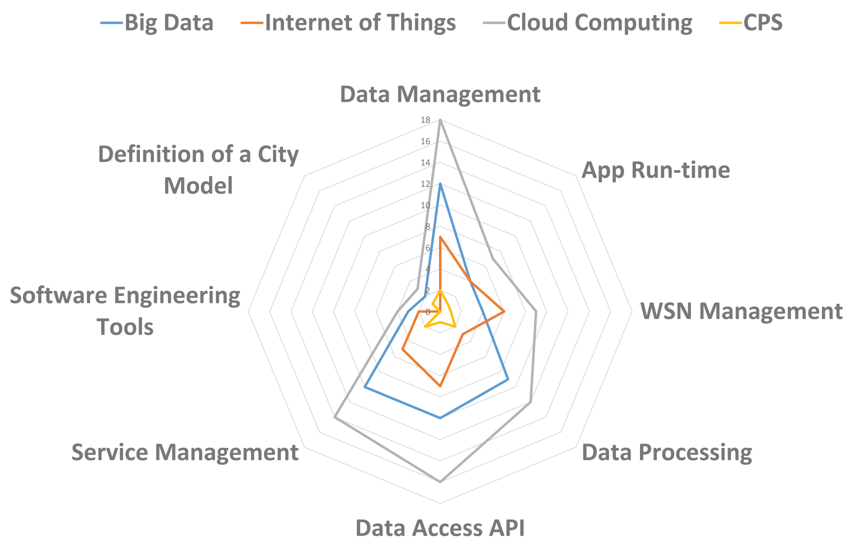

Fig. 9. Relationship between functional requirements and enabling technologies.

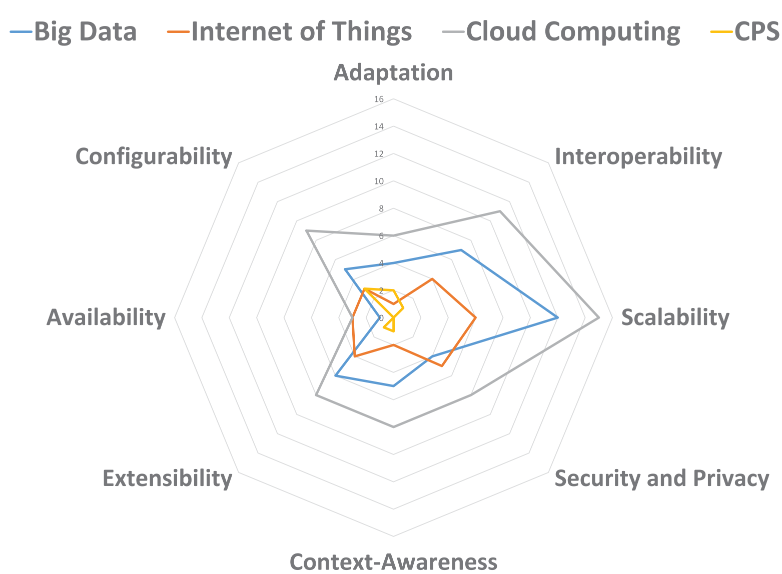

Fig. 10. Relationship between non-functional requirements and enabling technologies.

for most data-related activities, such as storing, analyzing, and sharing, and Cloud Computing provides a scalable and elastic environment for storing and processing city data.

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

Figure 10 illustrates non-functional requirements and 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 technologies. For example, all the CPS platforms handle configurability. Extensibility is primarily offered by platforms that use Big Data, while interoperability, primarily by platforms that use IoT.

6.2 Challenges and Open Research Problems

Most of the significant challenges and research problems in implementing a platform for Smart Cities relate to data management. The most cited problem in the literature is that of ensuring the privacy of user data, because of the amount of personal and critical data that a platform needs to handle; of special concern are 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 groups mentioned scalability as a problem, because it certainly will be a great challenge to support the many devices, users, data, and services in a large metropolis.

An important and understudied issue is how to create a generic platform to support different cities' requirements. Some literature focuses on a particular city's platform, 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 discussion of how these platforms' components could be adapted to fit cities of different sizes and characteristics.

6.3 Implications

This article discusses key requirements that software platforms for Smart Cities should handle. In this section, we discuss the potential implications of our findings for Smart City stakeholders, such as platform developers, application developers, city managers, system operators, end-users, and researchers.

The enabling technologies highlight the infrastructure needed to build Smart Cities, and city managers can use this information to make better investment decisions. Big Data and Cloud Computing, for instance, deal with an enormous volume of data storage and network infrastructure to access data and services. The city should be equipped with sensors, actuators, and Internet services to take full advantage of the Internet of Things and Cyber-Physical Systems. Application developers likewise should be aware of these technologies while building their products. From an educational point of view, these technologies should be part of the training of the next generation of software engineers.

The reference architecture presented in this article highlights functional and non-functional requirements that platforms and application developers should consider when developing software for Smart Cities. For platforms developers, our survey indicates they must find ways to effectively and efficiently deal with large-scale heterogeneous and distributed systems, as well as critical and personal data. For application developers, the reference architecture shows what kind of services and data they can use to implement applications for 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 routines, confronting issues such as urban mobility, air pollution, and heath care.

This survey can also help Smart City researchers by discussing open research questions and challenges towards supporting smarter cities, which can guide future research.

Smart City platforms will also impact *e-government*, *e-democracy*, and *e-participation*. Such platforms can leverage e-government services available to the public and integrate governmental services. Governmental service quality can also be improved with real-time data collected from the city ICT infrastructure, like the usage of city resources and facilities in domains such as health, education, water, energy, transportation, and waste.

E-democracy and e-participation can also be facilitated in their three main aspects: transparency, openness, and engagement (Van der Meer et al. 2014). Transparency deals with the ability to consult documents with information on city management decisions. Openness refers to the availability of data about city processes and infrastructure in addition to the concept of open data discussed in Section 2.2.3. Finally, engagement describes the opportunity for citizens to contribute to public decisions. Smart City platforms can, for example, enable city managers to share data about city decisions, processes, and infrastructure. E-participation can also be improved using tools that enable citizens to point out problems in the city, as already happens in Santander and Amsterdam. These tools can facilitate more factual, evidence-based, and transparent policy decision-making, promoting trust in the local government and satisfaction, as pointed out by Sivarajah et al. (2016) and Kim and Lee (2012). Therefore, Smart City platforms should reflect the emerging trends in the use and adoption of e-participation around the world (Hagen et al. 2015).

6.4 Limitations

In this survey, we described 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, which are used by only a few platforms or for a specific purpose. Thus, they are not reported as a fundamental architectural component of the platforms. There might be technologies, though not deemed relevant today (and thus not identified in this survey), that may become highly significant in the future.

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

Finally, we classified the articles according to the enabling technologies only when they were explicitly mentioned. However, we noticed that some articles highlighted them as a motivating aspect or future work. For example, 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 CONCLUSIONS

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

Using a software platform rather than *ad hoc* solutions is a more robust and sustainable way to support the features a Smart City environment needs. In this article, we surveyed the current research on Smart City Platforms, aiming to discover relevant requirements and ways to facilitate the development, integration, and deployment of Smart City applications. We analyzed 23 projects from different groups, proposing multiple approaches for the development of a software platform to answer our general research question: What characteristics should software platforms provide for enabling the construction of scalable integrated Smart City applications?

Based on the analyzed projects, we derived a unified reference architecture supporting the main requirements needed to build a Smart City software platform. Thus, this article 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 to the three research sub-questions. In answering RQ1 ("What are the enabling technologies used in state-of-the-art software platforms for Smart Cities?"), we identified that Internet of Things, Cloud Computing, Big Data, and Cyber-Physical Systems as the most cited enabling technologies. Answering RQ2 ("What requirements should a software platform for Smart Cities meet?"), we related 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 needs that should be addressed when using a specific enabling technology. In contrast, it helps to determine 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 in the literature. In this sense, an important contribution of this survey, especially for developers and researchers of software platforms, is identifying which platform components should be the focus of future work.

This survey described several Smart City initiatives still in their initial phases, which pose multiple challenges and open problems. A collaborative effort from research groups, commercial companies, NGOs, and governments is required to tackle the multitude of scientific, technical, political, and social problems related to establishing functionally Smart Cities and reaching the ultimate goal of improving the quality of life of all city citizens, irrespective of their social and financial situation.

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 *Proceedings of the 2014 11th International Bhurban Conference on Applied Sciences and Technology (IBCAST'14)*. IEEE, 414–419.

Eiman Al Nuaimi, Hind Al Neyadi, Nader Mohamed, and Jameela Al-Jaroodi. 2015. Applications of big data to smart cities. *J. Internet Serv. Appl.* 6, 1 (2015), 25.

Theodoros Anagnostopoulos, Kostas Kolomvatsos, Christos Anagnostopoulos, Arkady Zaslavsky, and Stathes Hadjiefthymiades. 2015. Assessing dynamic models for high priority waste collection in smart cities. *J. Syst. Softw.* 110 (2015), 178–192.

Giuseppe Anastasi, Maximiliano Antonelli, Alessio Bechini, Simone Brienza, Eleonora D'Andrea, Domenico De Guglielmo, Pietro Ducange, Beatrice Lazzerini, Francesco Marcelloni, and Armando Segatori. 2013. Urban and social sensing for sustainable mobility in smart cities. In *Proceedings of the Sustainable Internet and ICT for Sustainability (SustainIT'13), 2013*. IEEE, 1–4.

Leonidas G. Anthopoulos and Christopher G. Reddick. 2016. Understanding electronic government research and smart city: A framework and empirical evidence. *Inform. Polity* 21, 1 (2016), 99–117.

Wolfgang Apolinarski, Umer Iqbal, and Josiane Xavier Parreira. 2014. The GAMBAS middleware and SDK for smart city applications. In *Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM'14)*. IEEE, 117–122.

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. 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 *Proceedings of the 2011 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'11)*. IEEE, 229–234.

Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Comput. Netw.* 54, 15 (2010), 2787–2805.

Malcolm Bain. 2014. Sentilo–Sensor and Actuator Platform for smart Cities. 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 *Proceedings of the 2012 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST'12)*. 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 *Proceedings of the IIMC International Information Management Corporation Conference (eChallenges'11).*

Carolina Tripp Barba, Miguel Angel Mateos, Pablo Reganas Soto, Ahmad Mohamad Mezher, and Mónica Aguilar Igartua. 2012. Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights. In *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium (IV'12).* IEEE, 902–907.

Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. 2016. Integration of cloud computing and internet of things: A survey. *Future Gener. Comput. Syst.* 56 (2016), 684–700.

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 *Proceedings of the 2010 1st IEEE International Conference on Smart Grid Communications (SmartGridComm'10).* IEEE, 49–54.

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 *Proceedings of the 7th International Workshop on Semantic Sensor Networks (TC/SSN@ ISWC'14).* 51–66.

Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. 2011. Smart cities in europe. *J. Urban Technol.* 18, 2 (2011), 65–82.

Kate Carruthers. 2014. Internet of things and beyond: Cyber-physical systems. *IEEE IoT Newslett.* 2014 (2014).

Min Chen, Shiwen Mao, and Yunhao Liu. 2014. Big data: A survey. *Mobile Netw. Appl.* 19, 2 (2014), 171–209.

Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, and Ernoe Kovacs. 2015. Building a big data platform for smart cities: Experience and lessons from santander. In *2015 IEEE International Congress on Big Data (BigData Congress'15).* IEEE, 592–599.

Annalisa Cocchia. 2014. Smart and digital city: A systematic literature review. In *Smart City*, Renata Paola Dameriand Camille Rosenthal-Sabroux (Eds.). Springer International Publishing, 13–43.

Louise Coetzee and Johan Eksteen. 2011. The internet of things–promise for the future? An introduction. In *Proceedings of the International Sustainability Transitions Conference (IST'11)).* IEEE, 1–9.

Welington 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 *Proceedings of the 2014 International Conference on Collaboration Technologies and Systems (CTS'14).* IEEE, 104–112.

Salvatore Distefano, Giovanni Merlino, and Antonio Puliafito. 2012. Enabling the cloud of things. In *Proceedings of the 2012 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'12).* IEEE, 858–863.

Soufiene Djahel, Ronan Doolan, Gabriel-Miro Muntean, and John Murphy. 2015. A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Commun. Surv. Tutor.* 17, 1 (2015), 125–151.

Asma Elmangoush, Hakan Coskun, Sebastian Wahle, and Thomas Magedanz. 2013. Design aspects for a reference M2M communication platform for smart cities. In *Proceedings of the 2013 9th International Conference on Innovations in Information Technology (IIT'13).* IEEE, 204–209.

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 1st International Conference on IoT in Urban Space.* Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 1–6.

Marcio Zamboti Fortes, Vitor Hugo Ferreira, Guilherme Gonçalves Sotelo, Andressa S. Cabral, Weules Fernandes Correia, and Orestes L. Castaneda Pacheco. 2014. Deployment of smart metering in the Búzios city. In *Proceedings of the 2014 IEEE PES Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA'14).* IEEE, 1–6.

José Antonio Galache, Takuro Yonezawa, Levent Gurgen, Daniele Pavia, Marco Grella, and Hiroyuki Maeomichi. 2014. ClouT: Leveraging cloud computing techniques for improving management of massive iot data. In *Proceedings of the 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA'14).* IEEE, 324–327.

Rudolf Giffinger, Christian Fertner, Hans Kramar, Robert Kalasek, Nataša Pichler-Milanovic, and Evert Meijers. 2007. *Smart Cities-Ranking of European Medium-Sized Cities.* Rapport technique. Vienna Centre of Regional Science.

Sylva Girtelschmid, Matthias Steinbauer, Vikash Kumar, Anna Fensel, and Gabriele Kotsis. 2013. Big data in large scale intelligent smart city installations. In *Proceedings of the International Conference on Information Integration and Web-based Applications and Services.* ACM, 428–432.

Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* 29, 7 (2013), 1645–1660.

Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, and Xingshe Zhou. 2015. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *Comput. Surv.* 48, 1 (2015), 7:1–7:31.

Levent Gurgen, Ozan Gunalp, Yazid Benazzouz, and Mathieu Gallissot. 2013. Self-aware cyber-physical systems and applications in smart buildings and cities. In *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 1149–1154.

Loni Hagen, Jess Kropczynski, Catherine Dumas, Jisue Lee, Fatima Espinoza Vasquez, and Abebe Rorissa. 2015. Emerging trends in the use and adoption of E-participation around the world. *Proc. Assoc. Inf. Sci. Technol.* 52, 1 (2015), 1–4.

Robert E. Hall, B. Bowerman, J. Braverman, J. Taylor, H. Todosow, and U. Von Wimmersperg. 2000. The vision of a smart city. In *Proceedings of the 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 Martinez, Sara Izquierdo Barthelemy, and Mario González Fernández. 2014. Crowd density estimation for public transport vehicles. In *Proceedings of the International Conference on Extending Database Technology/International Conference on Database Theory (EDBT/ICDT'14)*. 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*. ACM, 9–13.

José M. Hernández-Muñoz, Jesús Bernat Vercher, Luis Muñoz, José A. Galache, Mirko Presser, Luis A Hernández Gómez, and Jan Pettersson. 2011. Smart cities at the forefront of the future internet. In *The Future Internet Assembly*. Vol. 6656. Springer, 447–462.

Hendrik Hielkema and Patrizia Hongisto. 2013. Developing the helsinki smart city: The role of competitions for open data applications. *J. Knowl. Econ.* 4, 2 (2013), 190–204.

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. *J. Syst. Softw.* 110 (2015), 253–263.

ISO/IEC. 2015. Smart cities Preliminary Report. (2015).

Valérie Issarny, Nikolaos Georgantas, Sara Hachem, Apostolos Zarras, Panos Vassiliadist, Marco Autili, Marco Aurélio Gerosa, and Amira Ben Hamida. 2011. Service-oriented middleware for the future internet: State of the art and research directions. *J. Internet Serv. Appl.* 2, 1 (2011), 23–45.

ITU-T. 2014. Smart Sustainable Cities: An Analysis of Definitions. Tech. Report. FG-SSC-10/2014, (2014).

Isam Janajreh, Liu Su, and Fathi Alan. 2013. Wind energy assessment: Masdar city case study. *Renew. Energ.* 52 (2013), 8–15.

Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. 2012. Benefits, adoption barriers and myths of open data and open government. *Inf. Syst. Manage.* 29, 4 (2012), 258–268.

G. Kakarontzas, L. Anthopoulos, D. Chatzakou, and A. Vakali. 2014. A conceptual enterprise architecture framework for smart cities: A survey based approach. In *Proceedings of the 2014 11th International Conference on e-Business (ICE-B'14)*. IEEE, 47–54.

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 *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE, 381–386.

Zaheer Khan, Ashiq Anjum, Kamran Soomro, and Muhammad Atif Tahir. 2015. Towards cloud based big data analytics for smart future cities. *J. Cloud Comput.* 4, 1 (2015), 2.

Jaeho Kim and Jang-Won Lee. 2014. OpenIoT: An open service framework for the internet of things. In *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT'14)*. IEEE, 89–93.

Soonhee Kim and Jooho Lee. 2012. E-participation, transparency, and trust in local government. *Publ. Admin. Rev.* 72, 6 (2012), 819–828.

Michinaga Kohno, Yoshihiro Masuyama, Nobuyuki Kato, and Akihiko Tobe. 2011. Hitachi's smart city solutions for new era of urban development. *Hitachi Rev.* 60, 2 (2011), 79.

Daniel Lathrop and Laurel Ruma. 2010. *Open Government: Collaboration, Transparency, and Participation in Practice*. O'Reilly Media, Inc.

Danh Le-Phuoc, Hoan Quoc Nguyen-Mau, Josiane Xavier Parreira, and Manfred Hauswirth. 2012. A middleware framework for scalable management of linked streams. *Web Semant.* 16 (2012), 42–51.

Jung Hoon Lee, Robert Phaal, and Sang-Ho Lee. 2013. An integrated service-device-technology roadmap for smart city development. *Technol. Forecast. Soc. Change* 80, 2 (2013), 286–306.

Yong Woo Lee and Seungwoo Rho. 2010. U-city portal for smart ubiquitous middleware. In *Proceedings of the 2010 The 12th International Conference on Advanced Communication Technology (ICACT'10)*, Vol. 1. IEEE, 609–613.

Pu Liu and Zhenghong Peng. 2013. China's smart city pilots: A progress report. *IEEE Computer* 47, 10 (2013), 72–81.

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. European Parliament (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 Kishor S. Trivedi. 2012. Combining cloud and sensors in a smart city environment. *EURASIP J. Wireless Commun. Netw.* 2012, 1 (2012), 247.

Jennifer C. Molloy. 2011. The open knowledge foundation: Open data means better science. *PLoS Biol.* 9, 12 (2011).

Georgios Mylonas, Evangelos Theodoridis, and Luis Muñoz. 2015. Integrating smartphones into the smartsantander infrastructure. *IEEE Internet Comput.* 19, 2 (2015), 48–56.

Milind Naphade, Guruduth Banavar, Colin Harrison, Jurij Paraszczak, and Robert Morris. 2011. Smarter cities and their innovation challenges. *Computer* 44, 6 (2011), 32–39.

Rocco Papa, Carmela Gargiulo, and Adriana Galderisi. 2013. Towards an urban planners' perspective on smart city. *TeMA J. Land Use, Mobil. Environ.* 6, 01 (2013), 5–17.

Anbu Parkavi and Nagarajan Vetrivelan. 2013. A smart citizen information system using hadoop: A case study. In *Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'13)*. IEEE, 1–3.

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. Emerg. Telecommun. Technol.* 25, 1 (2014), 81–93.

Daniel Pérez-González, Raimundo Díaz-Díaz, and others. 2015. Public services provided with ICT in the smart city environment: The case of spanish cities. *J. Univers. Comput. Sci.* 21, 2 (2015), 248–267.

Riccardo Petrolo, Valeria Loscri, and Nathalie Mitton. 2014. Towards a cloud of things smart city. *IEEE COMSOC MMTC E-Lett.* 9, 5 (2014), 44–48.

Giuseppe Piro, Ilaria Cianci, Luigi Alfredo Grieco, Gennaro Boggia, and Pietro Camarda. 2014. Information centric services in smart cities. *J. Syst. Softw.* 88 (2014), 169–188.

Gilles Privat, Mengxuan Zhao, and Laurent Lemke. 2014. Towards a shared software infrastructure for smart homes, smart buildings and smart cities. In *Proceedings of the International Workshop on Emerging Trends in the Engineering of Cyber-Physical Systems*.

Junping Qiu, Yanhui Song, and Siluo Yang. 2010. Digital integrated model of government resources under E-government environment. In *Proceedings of the 2010 International Conference on Internet Technology and Applications*. 1–4.

Aviva Rutkin. 2014. How data can save a city. *New Sci.* 224, 2990 (2014), 24–25.

Flora Salim and Usman Haque. 2015. Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things. *Int. J. Hum.-Comput. Stud.* 81 (2015), 31–48.

Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, and others. 2014. SmartSantander: IoT experimentation over a smart city testbed. *Comput. Netw.* 61 (2014), 217–238.

Eduardo Felipe Zambom Santana, Daniel Macêdo Bastista, Fabio Kon, and Dejan S Milojicic. 2016. SCSimulator: An open source, scalable smart city simulator. In *Proceedings of the 34th Brazilian Symposium on Computer Networks (SBRC'16)*. Salvador, Brazil.

Hans Schaffers, Nicos Komninos, Marc Pallot, Brigitte Trousse, Michael Nilsson, and Alvaro Oliveira. 2011. Smart cities and the future internet: Towards cooperation frameworks for open innovation. In *Proceedings of the Future Internet Assembly 2011: Achievements and Technological Promises*. Springer, Berlin, 431–446.

Amir Sinaeepourfard, Jordi Garcia, Xavier Masip-Bruin, Eva Marín-Tordera, Jordi Cirera, Glòria Grau, and Francesc Casaus. 2016. Estimating smart city sensors data generation. In *Proceedings of the 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'16)*. IEEE, 1–8.

Uthayasankar Sivarajah, Vishanth Weerakkody, Paul Waller, Habin Lee, Zahir Irani, Youngseok Choi, Rebecca Morgan, and Yuri Glikman. 2016. The role of e-participation and open data in evidence-based policy decision making in local government. *J. Organiz. Comput. Electron. Commerce* 26, 1–2 (2016), 64–79.

Martin Stephenson, Giusy Di Lorenzo, and Pol Mac Aonghusa. 2012. Open innovation portal: A collaborative platform for open city data sharing. In *Proceedings of the 2012 IEEE International Conference onPervasive Computing and Communications Workshops (PERCOM'12)*. IEEE, 522–524.

Kehua Su, Jie Li, and Hongbo Fu. 2011. Smart city and the applications. In *Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC'11)*. IEEE, 1028–1031.

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 *Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom'12)*. IEEE, 815–820.

Kenji Tei and Levent Gurgen. 2014. ClouT: Cloud of things for empowering the citizen clout in smart cities. In *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT'14)*. IEEE, 369–370.

Sean Thornton. 2013. Chicago's WindyGrid: Taking Situational Awareness to a New Level. 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. Retrieved December 13, 2015 from 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)*. ACM, 60:1–60:5.

Toni GLA Van der Meer, Dave Gelders, and Sabine Rotthier. 2014. E-democracy: Exploring the current stage of e-government. *J. Inf. Policy* 4 (2014), 489–506.

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. *IEEE Commun. Mag.* 51, 6 (2013), 128–134.

Félix J. Villanueva, Maria J. Santofimia, David Villa, Jesús Barba, and Juan Carlos Lopez. 2013. Civitas: The smart city middleware, from sensors to big data. In *Proceedings of the 2013 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'13)*. IEEE, 445–450.

Eleni I. Vlahogianni, Kostanstinos 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 *Proceedings of the 2012 IEEE 12th International Conference on Computer and Information Technology (CIT'12)*. IEEE, 895–900.

Kaiyu Wan, Danny Hughes, Ka Lok Man, and Tomas Krilavicius. 2010. Composition challenges and approaches for cyber physical systems. In *2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA'10)*. IEEE, 1–7.

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, 2 (2009), 1–17.

Rong Wenge, Xiong Zhang, Dave Cooper, Li Chao, and Sheng Hao. 2014. Smart city architecture: A technology guide for implementation and design challenges. *Chin. Commun.* 11, 3 (2014), 56–69.

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. *J. Internet Serv. Appl.* 1, 1 (2010), 45–56.

Chao Wu, David Birch, Dilshan Silva, Chun-Hsiang Lee, Orestis Tsinalis, and Yike Guo. 2014. Concinnity: A generic platform for big sensor data applications. *IEEE Cloud Comput.* 1, 2 (2014), 42–50.

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. *Sci. Chin. Inf. Sci.* 58, 10 (2015), 1–18.

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. *IEEE IofT J.* 1, 1 (2014), 22–32.

Received September 2016; revised May 2017; accepted July 2017