Adopting an Open Smart City Platform: A Survey

Achilleas Achilleos, Christos Markides, Andreas Konstantinidis, Ioannis Giorkas
Dept. of Electrical and Computer Engineering and Informatics
Frederick University, Cyprus
Nicosia, Cyprus
{com.aa, com.mc, com.ca, st011855}@frederick.ac.cy

Georgia M. Kapitsaki, Christos Mettouris, George A. Papadopoulos
Dept. of Computer Science
University of Cyprus
Nicosia, Cyprus
{gkapi, mettour, george}@cs.ucy.ac.cy

***Abstract*— Many municipalities adopt nowadays smart technologies offering smart services to their citizens towards the vision of a smart city. Different research works have revealed that the realisation of a smart city and its smart services creates a complex, multi-layered, and multi-dimensional environment with multiple stakeholders. In these surveys, technology requirements have been defined and several platforms were examined, and in most cases reference architectures have been proposed. However, these surveys mostly focus on the technology, while city governance is not effectively addressed via the requirements defined. In this work, we claim that the main goal should be to support digital transformation using technologies for developing smart, creative, innovative, and sustainable cities. Thus, eight open source platforms have been analysed against a set of governance-based and technology-driven requirements extracted and refined from earlier works. This survey provides a reference framework that enables policy makers, developers, researchers, and end-users to adopt a city governance strategy via an open source platform with the required technological features to realise a smart city research, investment, and innovation project.**

Keywords—smart city, governance, open source, platforms.

I. INTRODUCTION

While in many cases developing smart systems as individual city services provides solutions to citizen problems, still the potential is realised with appropriate city governance and the adoption of an integrative platform that satisfies key smart city requirements. As defined and emphasised in [1], the realisation of a Smart City is all about integrating software services and applications, so as to improve regular city services and thus ease and facilitate the life of citizens. Also creating independent vertical applications for each city domain is not sufficient for realising a Smart City environment [2].

Barcelona's Chief Technology Officer, Francesca Bria, states: "The problem of the smart city has been that when you start with technology without a strong idea of why you are deploying the technology and for what kind of needs, then you only end up solving technology problems" [2]. The aim of our work is fully aligned with the above statement, in that the necessary first step is to look at how technology can serve people in a smart city context. More importantly, successful Smart City initiatives strongly rely on advanced technologies innovation and governance, which are essential prerequisites for developing smart, creative, innovative, and sustainable cities [4].

In the case where a city begins with the implementation of vertical use cases (e.g., smart parking, smart street lighting, etc.) using different technologies and assigning critical urban services to big vendors, it immediately starts facing issues that are technology related, rather than addressing the citizens' needs and improving their standard of living. This creates an additional problem. By outsourcing critical urban services to big providers, the ability to shift from one provider to another is eliminated, while control of all the data, and ownership becomes ambiguous [2], thus resulting to the so-called vendor lock-in problem. This creates interoperability issues, which is considered the most critical requirement that has a direct and two-fold relation both with smart city technology, as well as city governance.

In fact, these views are shared in this work. The claim is that the two key points for realising the smart city vision are: (i) digital transformation, i.e., how city governance should go about to work with technology, which requires a fundamental change in the relationship between the public sector, private sector, and citizens, and (ii) adopting the appropriate open source platform that provides the technological features required to realise the city's requirements and goals of digital transformation, thus attaining smart city governance. The term Smart Governance refers to "*the intelligent use of ICT to improve decision-making through better collaboration among different stakeholders, including government and citizens, can be strongly related to government approaches*" [5], [6].

The key goal of this work is to perform a survey of open source platforms for smart cities that can offer not-only technological solutions, but most importantly, support smart city governance. Specifically, this work defines *a platform for Smart Cities* as: "*An open-source, integrated set of ICT technologies and tools that: (i) support policy makers, business managers, developers, and researchers in adopting a city governance strategy; and (ii) offers all the required technologies for realising a smart city open research, investment and innovation project*."

The main research question addressed in this work is:

- *What requirements and features need to be satisfied from an open source ICT platform for realising Smart City Governance and Implementation?*

The survey performed in this work aims to offer answers to the above question. It provides insights and guidelines on open source platforms that can be adopted, and which platforms offer the necessary ICT tools to drive digital transformation and governance of smart cities. In this paper, Section II performs a survey of platforms that are *open source*. A differentiating point

978-1-7281-0846-9/19/$31.00 ©2019 IEEE

with existing works is that this work aims to *provide guidelines for adopting an open source platform that can support smart city governance and implementation*. Finally, Section III presents the conclusions of this work.

II. Survey of Open Source Smart City Platforms

A. Analysis Using High-Level Adoption Criteria

The key contribution of this work is a survey of open source platforms for smart cities, to review whether and how these platforms satisfy the technologies, requirements, and features identified are summarised in this section.

Initially, our online investigation on existing "open source smart city platform" and "smart city service platform" resulted in identifying eight platforms. Here, it is important to note that our analysis excluded the CityOS platform from the results, since it was mostly an innovation laboratory with many outdated and incoherent resources, and included the #SmartME as another smart city platform realized at the city of Messina. Based on these results an additional search was performed in scientific repositories, namely Springer, IEEE Xplore, ScienceDirect Elsevier, and Google Scholar. The results were analysed and resulted in 15 research papers, which are either describing a specific platform or are survey papers [7], [8], [9] examining platforms for smart cities. The study and review of these papers did not reveal any other platforms that adhere to the critical requirement of being "open source". In order to characterize a platform as open source, we adopted the respective definition from the Open Source Initiative (OSI).

TABLE I. Platforms Adoption Characteristics

Platform	Open Source license	Documentation	Smart City Use Cases	Active
FIWARE	AGPL v3	Extensive	Vertical, Horizontal	Yes
OpenIoT	LGPL v3	Medium	Vertical	No
OpenRemote	AGPL v3	High	Vertical	Yes
Kaa IoT	Apache License v2	Medium	Not available	No
InterSCity	MPL v2.0	High	Vertical, Horizontal	Yes
Sofia4Cities	Apache License v2	Low	Horizontal	Yes
Sentilo	EUPL v1.1 & LGPL v3	High	Horizontal	Yes
#SmartME	Apache License v2	Medium	Horizontal	Yes

The initial comparison conducted in this work is based on four abstract characteristics defined in Table I. These characteristics serve as the baseline for concluding as to which platforms will be included in the analytical comparison against the requirements and features proposed in this work. Since all platforms are open source carrying open source licenses, the primary criterion is for the platform to have an active community and a roadmap with currently ongoing activities. Moreover, the platform must have been applied in a horizontal way in a smart city context and have at least a medium number of resources that make it suitable for adoption, where horizontal integration represents the application of multiple smart city use cases (e.g. smart tourism, smart parking, etc.), and vertical integration indicates the application of single use case on a platform. In addition, the Documentation characteristic values are set on a scale from Low to Extensive with values: Low, Medium, High, and Extensive, where Low represents a minimal number of resources, Medium represents an adequate number of resources, High represents a large number of resources, and Extensive represents a large number of resources including access to an online academy and webinars respectively, in terms of availability of online documentation.

Thus, the Kaa IoT and the Sofia4Cities platforms have been excluded, as smart city use cases are not available, and the existing documentation for the platform was scarce in order to allow a deep analysis, respectively. Based on these criteria, the platforms that will be considered in the comparative analysis are: FIWARE, Sentilo, InterSCity, and #SmartME.

In Table II, a synopsis of the requirements and features extracted and refined from existing works is presented and our contribution focuses on identifying, and defining platform features that can explicitly support city governance, e.g., multi-tenancy [10], providers management, in a smart city context.

TABLE II. Platform Requirements and Features

Requirement/ Feature	Requirements and Features Description
Interoperability	Services (*service management*) and wireless sensor network devices (*WSN Management*) should communicate with each other, be accessible from any device and platform (*accessibility*) and from any location (*mobility*).
Adoption Model	Capability to deploy the smart city platform on the preferred infrastructure (*architecture – F1*), choice to deploy their services and applications on the city's platform or their own infrastructure (*runtime environment – F2*), and ability to ensure minimal interruption for their clients by deploying on city infrastructure or at their premises based on their needs (*reliability – F3 and availability – F4*).
Stakeholders Management	Institutional organisation via the hierarchical management of stakeholders in the smart city platform (*multi-tenancy – F5*), ability for stakeholders to manage their own providers, e.g., sensor, data (*providers management – F6*), and support for Authentication, Authorisation and Accounting (AAA – F7), which allows controlling access to resources, enforcing policies, auditing and billing for services (*security*).
Data Management	Define access to resources (*data access/privacy – F8*), support in-memory and on-disk storage for both static and real-time data (*data storage - F9*), transform and store data in different formats to exploit simple visualization tools (*data warehousing & visualization – F10*) and the capability to gain meaningful insights and make decisions by using big data and AI tools (*data processing and analytics – F11*).

TABLE III. PLATFORMS ADOPTION CHARACTERISTICS

Platform	Interoperability	Adoption Model				Stakeholders Management			Data Management			
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
FIWARE	FS	√	Local Server, SaaS	√	√	√	×	√	√	Orion Context Broker – MongoDB	Cygnus GE – PostgreSQL, MySQL, MongoDB, AWS DynamoDB	Cygnus GE – Hadoop, Storm, Spark or Flink
InterSCity	PS	×	Local Server	√	√	×	×	×	×	RabbitMQ, PostgreSQL, MongoDB, Redis	×	×
Sentilo	FS	√	Local Server, VM, SaaS	√	√	√	√	√	√	MongoDB, Redis	MySQL, openTSDB	Elasticsearch, Kibana
#SmartME	PS	×	Local Server	√	√	×	×	×	×	MySQL, WAMP protocol	×	×

B. COMPARATIVE ANALYSIS

In this section a detailed comparative analysis of the four platforms in terms of the requirements and features identified is presented. In terms of the interoperability requirement, three scales are defined: fully-support (FS), partial-support (PS), and no support (NS). The FS scale value requires for a platform to provide (i) high-level services that can be used by different services and applications provided by various stakeholders in the city offering secure access to those services, as well as (ii) an API management tool that allows API discovery, testing, etc. The PS scale value refers to one of the two being supported, while if none of the two requirements is supported then the scale is designated as NS. A summary of the analysis can be found in Table III, where the features F1-F11 are briefly outlined in Table II, and explained in detail in this section.

The FIWARE platform offers the FIWARE NGSI (Next Generation Service Interface) API that provides high-level services and enables different stakeholders in a city to develop services and applications for the cities [11]. Specifically, a data model for representing context information based on the notion of entities, where a context entity is a “Thing”, e.g., a sensor, actuator that can be created, updated, and deleted. There is also a context data interface for exchanging information with entities, and a context availability interface for exchanging information on how to obtain context information. These APIs enable service management, WSN management, mobility and accessibility of these services and devices from any provider that requires them to create applications and services. Finally, HTTPS access is provided to different stakeholders/tenants using authentication tokens, as well as the APInf tool can be used for API management [12].

The InterSCity platform provides a Resource Adaptor API, which is a service that can encapsulate resources. A resource is a “Thing” in the real world, which can be a sensor, an actuator, or a microcontroller with many sensors and actuators. Moreover, Resource Catalog and Data Collector APIs are offered. The catalog service provides data, such as status, configuration, location, and identification for each registered resource in the platform, while the data service is used both by applications and the catalog service, and offers developers access to information coming from devices and services (i.e., virtual sensors) scattered in the city. The Actuator Controller API enables sending commands to actuators. In addition, the Discovery Service API provides methods to find resources UUIDs collected from the Resource Catalog API. The platform offers high-level services for service and WSN management, which enable external services and applications to access and manipulate resources from any location in the city, providing quality city-scale services [13]. The platform does not enforce any security scheme for API calls, but it allows using the Kong Dashboard to manage the InterSCity's APIs and using Swagger to view, test, and learn about the APIs.

The Sentilo platform offers a high-level API that allows users/stakeholders to externally interact with the smart city platform. It defines the Security service that enables API authentication, while the Catalog service allows registering clients (applications/modules and providers/sensors) in the platform. The Subscription service enables clients to subscribe to system events, such as data, orders, and alarms. The Data service allows reading, writing, or deleting the observations of the registered sensors, while the alarm service allows to record and retrieve alarms associated with an alert stored in the system catalog. The order service allows to send or retrieve orders from sensors/actuators [14]. The platform provides HTTPS access to different stakeholders and providers using authentication tokens. Although the platform does not offer a tool for API management, it can be integrated with tools such as Swagger.

The #SmartME platform provides the IoTronic Services, it integrates Swagger to view, test, and learn about the available APIs, However, the platform only provides API access through the token/account of the administrator of the #SmartME platform [15]. The IoTronic Services include the

Authentication API for enabling security when calling the platform resources and services. The platform also includes the Boards and Services APIs that enable management of the devices (e.g., microcontrollers, sensors, actuators), access to services offered by the boards. The platform also offers the Projects and Users API for managing projects and users and the Virtual Networks API for creating and managing virtual networks among platform objects [15], [16].

In terms of the runtime deployment feature (F1) of the adoption model, the FIWARE platform enables via the Docker Generic Enabler to offer basic docker container hosting capabilities so that developers can create and deploy their own services [12], [17]. The InterSCity platform does not provide any explicit deployment and runtime environment for running smart city applications, while Sentilo enables developers to deploy their applications on the platform by using the Apache Tomcat server. The SmartME# platform does not offer a runtime environment. FIWARE and Sentilo do not enforce the use the runtime environment available via the platform, however if an organisation (e.g., startup wanting to reduce costs) wishes to, it can deploy the application on top of the platform.

The architecture model (F2) refers to the setup process and the availability of alternative methods to use, test, and adopt each platform. FIWARE's easiest evaluation method is to use the cloud lab, which provides a working instance of FIWARE available for experimentation, which requires registering and logging in to use the running demonstrator instance of the platform. Installing the platform using docker images on a local server is complex and time consuming, as it requires to setup, configure, and integrate each platform component separately [12]. The InterSCity platform provides docker containers/images that can be used to easily and quickly setup the platform in GNU Linux Debian 8 and 9 environments by executing the available installation scripts on a local server [18], [19]. The scripts prove useful, for testing and making a decision whether to adopt the platform, but a VM or SaaS running instance of the platform will be even more appropriate.

Sentilo provides the option to download a VM than can be used directly to test the platform and its capabilities. There are SaaS providers of the Sentilo platform, as a second adoption strategy. Available documentation covers manual installation, configuration, testing and using the platform on a local server [14]. The #SmartME project provides documentation on GitHub with details on the installation procedure of the platform as a standalone version on Ubuntu 14.04, 16.04 and Debian 9 or as a resource management service on the OpenStack cloud [15].

Following up with the sustainability/maintainability (F3) and availability/recoverability (F4) features it can be claimed that these are satisfied in all cases. In fact, city stakeholders can exploit the technical capabilities and human resources of the partner that will adopt the platform in the smart city, so as to deploy their own applications and services. At the same time, all platforms can be deployed on a cloud provider such as Azure, Amazon EC2, OpenStack, to exploit the benefits provided by cloud computing, and the desired QoS levels.

The multi-tenancy [10] feature (F5) is a method to organise access to the resources available in a smart city. It is supported in FIWARE by the Orion Context Broker, which implements a multitenant/multiservice model based on logical database separation. The multiservice ensures that the entities, attributes, and subscriptions of one service/tenant are hidden from other services/tenants, based on the policies defined. The Sentilo platform also supports multi-tenancy. Sentilo provides the capability of creating and managing virtual instances/profiles related with different organizations (e.g. SMEs, public services). The different virtual instances provide each organisation with their own administration dashboard and public portal, even with a different style, its own context, entities and data, as well as the ability to share data in accordance to its requirements. The InterSCity and #SmartME projects do not support the multi-tenancy feature, although such a logical separation could be achieved using the projects' and users' concepts.

Providers management (F6) can be considered as a related feature to multi-tenancy. Specifically, a tenant (e.g., SME) of a smart city platform can have multiple providers (e.g., departments), each having their own distinct components, sensors, actuators, and applications. Therefore, it is required for platforms to adhere to this feature, so as to promote the required institutional organisation in the context of the smart city. The survey revealed, to the best of our knowledge that only the Sentilo platform satisfies this feature. It may be possible to accomplish this in FIWARE, by creating a tenant and exploiting the logical database separation, but this does not provide the hierarchical organisational structure of a smart city as envisioned in this work.

The AAA security feature (F7) is important in order to avoid unauthorized access to smart city resources, as well as to enforce policies, auditing usage, and providing the information necessary for billing services. Keyrock is the FIWARE component responsible for Identity Management, which enables (in conjunction with other security components such as PEP Proxy and Authzforce) to add OAuth2 authentication, authorisation and accounting security to the smart city platform services and applications. The Sentilo platform also enforces the AAA security scheme. Finally, apart from InterSCity, the APIs and dashboards of the other platforms can be accessed using an HTTPS connection for increased security. The #SmartME project has the limitation that only the token/account of the administrator can be used of API calls, something that hinders interoperability. Finally, the security scheme enforced by FIWARE and Sentilo enables the different tenants of the platforms to specify the data access level as public or private, in order to guarantee data privacy, and authorised accessibility (F8).

FIWARE adopts and supports a wide variety of options in terms of data storage and data warehousing (F9, F10). The Orion Context Broker generic enabler (i.e., component) enables developers to manage the entire lifecycle of context information including updates, queries, registrations and subscriptions [11]. The Orion Context Broker is an implementation of the FIWARE Publish/Subscribe Context Broker Generic Enabler [12]. Context information is stored in MongoDB, but the Short Time Historic (STH) Comet generic enabler of FIWARE allows managing (storing and retrieving) historical raw and aggregated time series information about the evolution in time of context data registered in the Orion Context Broker instance. Also, Cygnus is a generic enabler that allows persisting certain sources of data in certain configured third-party storages, creating a historical view of such data, for managing the history of context that is created as a stream of data (F11). The data can be injected into multiple data stores, such as PostgreSQL, MySQL, MongoDB or AWS DynamoDB, as well as Big Data platforms, such as Apache Hadoop, Storm, Spark or Flink. The platform also offers a number of generic enablers to process, analyse or visualize context information for the purpose of implementing smart city services and applications. FIWARE widely supports all data management features, besides a processing engine that allows using AI algorithms. Open source machine learning frameworks such as Tensorflow or Keras can be integrated with the platform.

Initially, the InterSCity platform adopted PostgreSQL in all microservices, since it supports georeferenced queries, which are important in the smart city domain [18]. Currently, the InterSCity platform supports and performs asynchronous messaging by using RabbitMQ, a widely used, lightweight, open source messaging middleware that implements the AMQP protocol. The PostgreSQL database is used for the Resource Catalog and Adaptor microservices, and Redis was used as cache for both services (F9). For the Data Collector and Actuator Controller microservices, e.g., to access data collected from Resources, the input obtained from sensors is stored dynamically in the Percona memory engine, and data is stored using MongoDB [18]. The Data Collector also uses database caching supported by Redis to provide low-latency readings of the latest data collected by city resources [18]. Consequently, the platform does not explicitly support data warehousing (F10), but it utilises both PostgreSQL and MongoDB. InterSCity does not support data visualization and data processing and analytics using specific open source tools (F11). In fact, this aspect is covered in the implementation of each smart city use case implemented on top of the platform, such as the smart parking application [13], [18].

The Sentilo platform allows users to publish and retrieve data, e.g., from sensors, applications, and to subscribe and be informed on system events [21], [22]. This functionality is implemented as a Java component that uses the Redis server as the publication-subscription platform [14]. Real-time, in-memory data storage is proportional to Redis deployment and depends on the amount of physical memory available for the Redis server. Mongo-DB is used for on-disk permanent data storage, which allows to delete in-memory data based on the settings configured for the Redis server, e.g., in the Barcelona deployment, the data is deleted after approximately one week [14]. The Sentilo platform also supports the notion of data processing agents, which enables alerts, and supports data warehousing, visualization, processing and analytics tasks (F9 – F11). These agents and integration methods are optional and can be enabled, if required, and it allows exporting the published events and historical platform data to a relational database (i.e., MySQL) using the Relational Database Agent (RDA) [14], [21]. The Historian Agent (HA) allows uploading historical data and events to OpenTSDB and the Activity Monitor Agent (AMA) can be used for uploading historical data and events to Elasticsearch. The HA enables handling large volumes of data produced in a smart city, by using OpenTSDB that provides a scalable solution for time series data [14]. OpenTSDB installs of top of Hadoop Distributed File System (HDFS), and Hbase, and exposes a REST API, which can be used from the Grafana open platform for data analytics and monitoring. The AMA supports storing and exploiting historical events and data, through Elasticsearch, and Kibana, for data visualization [14]. Logstash can be used for collecting logging information (e.g., login errors, invalid messages). Thus, the intelligent analytics aspect via artificial intelligence methods and tools is the only feature not currently covered by the Sentilo platform, but open source AI frameworks can be integrated (F11).

Finally, the #SmartMe project supports the data storage requirement (F9). In real-time data using WAMP, which adds the higher-level messaging patterns of remote procedure call (RPC) and Pub/Sub to the WebSocket protocol. It also supports storing the IoTronic database that includes the catalog information (e.g., users, boards, sensors) and data observations (e.g., temperature sensor observations) in a MySQL database. To the best knowledge of the authors, the features of data warehousing and visualization and data processing and analytics are not supported by the platform (F10 – F11).

III. Conclusions and Future Work

The survey performed aims to provide a detailed examination and analysis of the technical features and requirements covered by existing open source smart city platforms and identify any missing capabilities. It does not aim to select the best open source platform, but it rather enables policy makers, developers, researchers and end-users to adopt an open source platform that provides the required technological features to realise the smart city strategy, enables governance as part of a research, investment and/or innovation project. In this context, the four platforms that were examined in detail in Section II provide many of the features identified in this work.

The FIWARE platform is the platform that provides all necessary components (i.e., generic enablers as they are referred to in the FIWARE terminology) and covers the entire set of features for the smart city realisation. The documentation is extensive, and the platform has been already applied successfully in more than 100 cities. On the downside, due to the many components available that are a result of different projects, the configuration and integration of those generic enablers in a holistic platform requires heavy investments in terms of time, effort, and human resources. Advanced knowledge and study of each FIWARE component is required before adopting the smart city platform that horizontally delivers the services and applications needed to its citizens. On the other hand, a variety of technological choices are available, and integration with different Big Data platforms, provides greater flexibility. In the context of a large city and the high availability of human and other resources to invest then FIWARE is indeed an appropriate choice.

The Sentilo platform also provides direct support for most of the features identified in this work and has been applied massively in the context of the Smart City of Barcelona. It also provides integrations to external tools, through the agents that are implemented in the platform, which enables support of the additional required features. The documentation is also excellent and provides the necessary resources for both understanding the operation and technical details of the platform, and it also provides easy to follow guidelines to setup the platform. Its greatest advantage is the straightforward method to setup, test and adopt an integrated version of the platform, which requires moderate investment in terms of human and other resources. Sentilo can be an optimal choice for a small city that has limited resources to invest in the adoption of a smart city platform.

The InterSCity and #SmartMe are also delivering support for most of the features and requirements identified in this work. Improvements can be made in the context of smart city governance that is directly linked in this work to the features of multi-tenancy, providers management and security. InterSCity and #SmartMe do not support multi-tenancy and providers management, which means that institutional organisation within the context of a smart city cannot be enforced. Moreover, in terms of security, HTTPS is only supported by #SmartME. Features such as services accounting is not supported by any of the two platforms for billing services.

The Sentilo platform has been adopted and is currently deployed and used at the Mobile Devices Laboratory (MDL) of the Frederick University, Cyprus, with the aim to pave the way and promote research, and innovation in the field of Smart Cities in Cyprus, which is currently lacking behind mainly due to governance issues. The use cases include a Smart Incident Reporting System (SIRS), which enables citizens to report real-life incidents experienced in their city, and a Smart Parking system that detects parking availability in real-time. Future work aims to propose a smart city governance model, driven on the basis of the results in this survey, and to engage in a dialogue with stakeholders in Cyprus to utilise the adopted platform in a smart city research and innovation project.

ACKNOWLEDGMENT

The work is partly funded by the NET-SMART project, ENCOURAGEMENT-H2020/0916 programme of the local Research Promotion Foundation (RPF), Cyprus.

REFERENCES

[1] D. Washburn, U. Sindhu, S. Balaouras, R.l A. Dines, N. M. Hayes, and L. E. Nelson, “Helping CIOs understand smart city initiatives”. Forrester Research, Growth 17, 2009.

[2] R. M. Kanter and S. S. Litow, “Informed and Interconnected: A Manifesto for Smarter Cities”, Working Paper 09-141, March 2009, Faculty Unit(s): General Management, Harvard Business School, Last Accessed June 28, 2019. Available: https://hbswk.hbs.edu/item/informed-and-interconnected-a-manifesto-for-smarter-cities

[3] R. Forster, “How Barcelona’s smart city strategy is giving ‘power to the people’”, Smart Cities, April 3, 2018, Last Accessed June 28, 2019. Available: https://news.itu.int/how-barcelonas-smart-city-strategy-is-giving-power-to-the-people/

[4] N. V. Lopes, “Smart governance: A key factor for smart cities implementation”, 2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC), Singapore, 2017, pp. 277-282, doi: 10.1109/ICSGSC.2017.8038591.

[5] E. Almirall, J. Wareham, C. Ratti, P. Conesa, F. Bria, A. Gaviria, A. Edmondson, “Smart Cities at the Crossroads: New Tensions in City Transformation”, California Management Review, 59, 2016, 141-152. 10.1177/0008125616683949.

[6] E. Ferro, B. Caroleo, M. Leo, M. Osella and E. Pautasso, “The Role of ICT in Smart Cities Governance”, in: CeDEM13 Conference for E-Democracy and Open Government, P. Parycek and N. Edelmann, eds., Danube University, Krems, Austria, 2013, pp. 133-145.

[7] E. Santana, A. Chaves, M. Gerosa, F. Kon, and D. S. Milojicic, “Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture”, ACM Journal of Computing Surveys, 50, 6, Article 78, November 2017, 37 pages. DOI: https://doi.org/10.1145/3124391.

[8] G. Kakarontzas, L. Anthopoulos, D. Chatzakou and A. Vakali, "A conceptual enterprise architecture framework for smart cities: A survey based approach," 2014 11th International Conference on e-Business (ICE-B), Vienna, 2014, pp. 47-54.

[9] W. M. da Silva, A. Alvaro, G. Tomas, R. Afonso, K. Dias, and V. Garcia, “Smart cities software architectures: a survey”, In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC), 2013 ACM, New York, NY, USA, 1722-1727. DOI: https://doi.org/10.1145/2480362.2480688.

[10] S. Cherrier, Z. Movahedi and Y. M. Ghamri-Doudane, "Multi-tenancy in decentralised IoT," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, 2015, pp. 256-261, doi: 10.1109/WF-IoT.2015.7389062.

[11] P. Salhofer, “Evaluating the FIWARE Platform: A Case-Study on Implementing Smart Application with FIWARE”, in Proceedings of the 51st Hawaii International Conference on System Sciences, 2018.

[12] FIWARE Foundation, e.V., “FIWARE Catalogue”, Last Accessed June 28, 2019. Available: https://www.fiware.org/developers/catalogue/

[13] A. M. Del Esposte, F. Kon, F. M. Costa, and N. Lago, “InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities”, in Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems, 2017, Volume 1: SMARTGREENS, ISBN 978-989-758-241-7, pages 35-46. DOI: 10.5220/0006306200350046.

[14] Barcelona City Council, “Sentilo Documentation – Release 1.8.0”, Last Accessed June 29, 2019. Available: https://buildmedia.readthedocs.org/media/pdf/sentilo-docs/latest/sentilo-docs.pdf.

[15] Stack4Things, GitHub Wiki, Last Accessed June 29, 2019. Available: https://github.com/MDSLab/stack4things.

[16] D. Bruneo, S. Distefano, F. Longo, G. Merlino, A. Puliafito, “Turning Messina into a Smart City: The #SmartME Experience”, In: A. Stratigea, E. Kyriakides and C. Nicolaides (eds), Smart Cities in the Mediterranean. Progress in IS. Springer, Cham, 2017.

[17] V. Araujo, K. Mitra, S. Saguna, C. Åhlund, “Performance evaluation of FIWARE: A cloud-based IoT platform for smart cities”, Journal of Parallel and Distributed Computing, Volume 132, 2019, 250-261, ISSN: 0743-7315, https://doi.org/10.1016/j.jpdc.2018.12.010.

[18] InterSCity, GitLab Wiki, Last Accessed June 29, 2019. Available: https://gitlab.com/interscity/interscity-platform/docs.

[19] A. de M. Del Esposte, E. F. Z. Santana, L. Kanashiro, F. M. Costa, K. R. Braghetto, N. Lago, F. Kon, “Design and evaluation of a scalable smart city software platform with large-scale simulations”, Elsevier Journal of Future Generation Computer Systems, Volume 93, 2019, 427-441, ISSN: 0167-739X, https://doi.org/10.1016/j.future.2018.10.026.

[20] F. Longo, D. Bruneo, S. Distefano, et al., “Stack4Things: a sensing-and-actuation-as-a-service framework for IoT and cloud integration”, Journal of Annals of Telecommunications, February 2017, Volume 72, Issue 1–2, pp 53–70. https://doi.org/10.1007/s12243-016-0528-5.

[21] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, E. Marín-Tordera, J. Cirera, G. Grau and F. Casaus, “Estimating Smart City Sensors Data Generation Current and Future Data in the City of Barcelona”, In Proceedings of Conference: The 15th IFIP Annual Mediterranean Ad Hoc Networking Workshop, June 2016.

[22] M. Bain, “Sentilo Case Study”, BGMA, European Commission (EC) Joinup, Last Accessed June 29, 2019. Available: https://joinup.ec.europa.eu/sites/default/files/document/2014-06/SENTILO%20case_joinup_v_1%202.pdf.