



HAL
open science

The discovery of “relevant” data-sources in a Smart City environment

Riccardo Petrolo, Salvatore Guzzo Bonifacio, Valeria Loscrì, Nathalie Mitton

► To cite this version:

Riccardo Petrolo, Salvatore Guzzo Bonifacio, Valeria Loscrì, Nathalie Mitton. The discovery of “relevant” data-sources in a Smart City environment. Proceedings of SSC - 2nd International IEEE SMARTCOMP Workshop on Sensors and Smart Cities , May 2016, St. Louis, Missouri, United States. hal-01300335

HAL Id: hal-01300335

<https://inria.hal.science/hal-01300335>

Submitted on 19 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The discovery of “*relevant*” data-sources in a Smart City environment

Riccardo Petrolo, Salvatore Guzzo Bonifacio, Valeria Loscri, and Nathalie Mitton
Inria Lille - Nord Europe, France. e-mail: {name.lastname@inria.fr}

Abstract—Among all the Internet of Things (IoT) applications, the Smart City concept has received significant attention in the last few years. The main motivation behind this interest is attributable to population growth and urbanization trend. Cities need, indeed, to be ready to face new challenges - e.g., traffic congestion, waste management, etc. - caused by this new amount of citizens. To address those issues, many IoT solutions have already been proposed and many others are still under investigation; anyway, those initiatives are all based on different standards and protocols, while the Smart City concept requires integration among all the stakeholders. In this paper, we present the VITAL architecture, which aims to integrate Internet-Connected Objects (ICOs) among multiple IoT platforms and ecosystems. In particular, we introduce the “ICOs and Services Discovery” module, which makes completely transparent, for users, the exploration of data-sources that are appropriate for his/her business context. This mechanism is at the basis of the Cloud of Things paradigm and a key feature as the platform agnostic property is an essential goal for VITAL.

Index Terms—Smart Cities, Internet of Things, Cloud of Things, Discovery.

I. INTRODUCTION

The Internet of Things (IoT) is signing a revolution, in both academic and industry fields, due to its capability to provide *connectivity for anything, from anywhere and anytime to anyone* [1]. The number of possible applications is massive and it effects desperate domains; to name a few, agriculture, logistic, health, environment, and smart city.

Among all those domains, in the last few years, the Smart City concept has received a significant research effort and technological development. The main motivation behind this massive interest is attributable to the population growth and the urbanization trend. With the beginning of the urbanization phenomena (late 18th century) cities started, indeed, to grow quickly; people moved from rural to urban areas in order to access major opportunities for jobs, education, housing, and transportation [2]. In 2010, 50% of the world’s population was living in cities. Following this trend, in 2050, 70% of the world’s population will be urban [3]; therefore, cities must be ready to face new challenges (e.g., environmental monitoring, traffic jam, waste management, etc.) caused by this new amount of citizens.

Although there is not yet a formal and universally accepted definition of Smart City, in [4] we delineate the concept by highlighting the need of *integration* and *interoperability* between all the city’s stakeholders. In this context, the IoT - which provides the devices needed to monitor and control a Smart City - represents a “*bad example*”. Indeed, despite

several initiatives have been proposed and many others are still under investigation, they are all based on different standards, protocols, and platforms. Those IoT ecosystems are, therefore, not always able to communicate between them, going against the above Smart City requirement.

To bridge this gap, the recent advances in IoT architectures together with the use of semantic web technologies - notably IoT ontologies -, are recognized as solutions [5] to enable the convergence of those silos and therefore to facilitate the development of added-value integrated cross-context applications [6]. In this paper, we present the European FP7 VITAL¹ project, an IoT architecture that enables the development, deployment, and operation of semantically interoperable applications for Smart Cities. In particular we introduce the “ICOs and Services Discovery” (SD) module, which deals with the exploration of appropriate data-sources for a specific business context. When a request is sent, the Discovery selects, among all the ICOs belonging to different IoT platforms, the ones that better suit the needs of the user; therefore, he/she can easily use the service/s offered by the ICO (e.g., temperature), without considering the underlying protocols and standards. This process is compliant with the Sensing-as-a-Service model that is at the basis of the Cloud of Things paradigm that we introduced in [4].

The paper is structured as follows: Section II introduces the VITAL architecture and how it is used in the Smart City context. In Section III we describe, in detail, the “ICO and Services Discovery” module of VITAL. Literature regarding IoT platforms and discovery of resources is presented in Section IV, while Section V concludes the paper.

II. VITAL

One of the most important objectives of VITAL is about the integration of Internet-Connected Objects (ICOs) among multiple IoT platforms and ecosystems. A very key factor, in this context, is represented by the virtualization of interfaces in combination with cross-context tools that enable the access and management of heterogeneous objects supported by different platforms and managed by different administrative stakeholders.

As shown in Figure 1, the VITAL architecture is organized in three main layers: *IoT Platforms and Data Sources*, *Platform Agnostic Management, Monitoring and Governance*,

¹<http://vital-iot.eu>

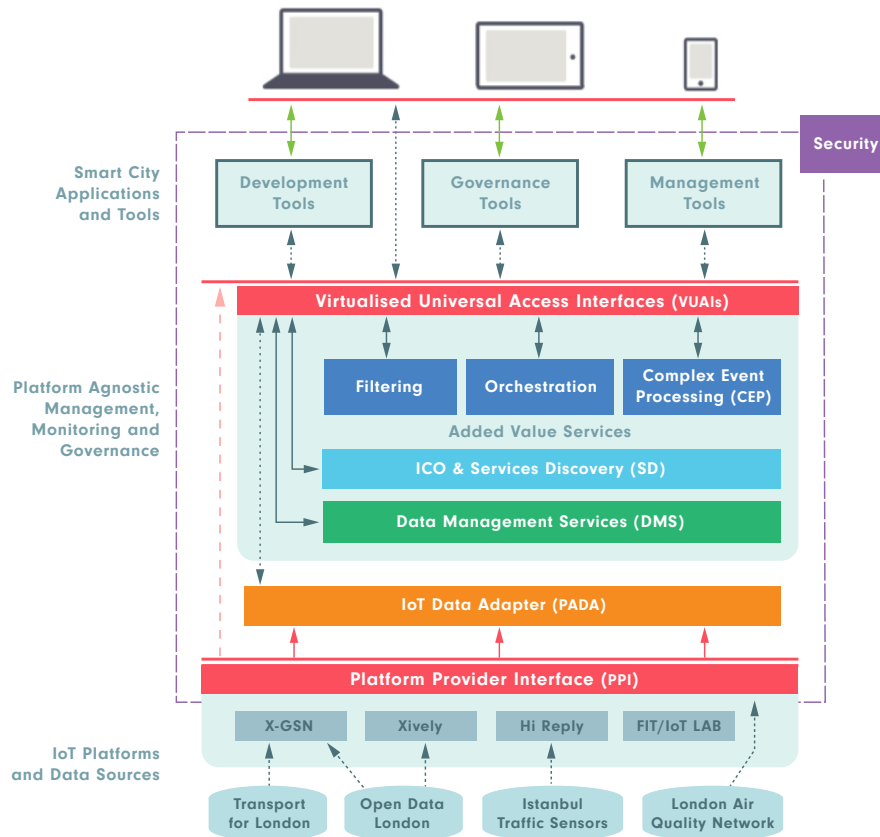


Fig. 1: The VITAL architecture.

and *Smart City Applications and Tools*. In the following, we present the features of fundamental modules:

- **IoT Platforms and Data Sources.** It is the first layer of the architecture, where different data-sources stand. In order to be virtualized and integrated into VITAL, those systems have to expose a well defined PPI (Platform Provider Interface).
- **Platforms Access and Data Acquisition (PADA).** The role of this layer is to access the low-level capabilities of the IoT Systems (through PPI) and to transform the acquired data and meta-data into a common data model (i.e., VITAL ontology).
- **Data Management Services (DMS).** This layer provides cloud-based functionality for managing data and meta-data. The offered services include data and meta-data persistence, creation of new data, and more. The DMS communicates, via REST interfaces, directly with PADA, Added Value Services and the VUAs (Virtualized Unified Access Interfaces).
- **ICOs and Services Discovery (SD).** This layer provides the means for discovering ICOs in the scope of horizontal integrated IoT applications spanning multiple platforms and business contexts. It directly interacts with the DMS in order to discover the “appropriate” resources for a

particular business context.

- **Added Value Functionalities.** It involves a set of complete services and tools:
 - *Filtering.* It provides the means for reducing the information associated with individual data streams persisted in the platform agnostic data management layer. Therefore, it reduces unwanted information, thereby optimizing processing performance and economizing on network bandwidth.
 - *Complex Event Processing (CEP).* It enables the processing of data-streams for multiple sources in order to identify patterns and/or infer events.
 - *Orchestration.* Its role is to combine and manage multiple services from the above-listed modules, in order to deliver new added-value services.
- **Smart City Applications and Tools.** At this layer, we support the development, integration, deployment, and operation of Smart City applications. This goes for instance with a complete environment to assist in the easy deployment and development of Smart City application [6].

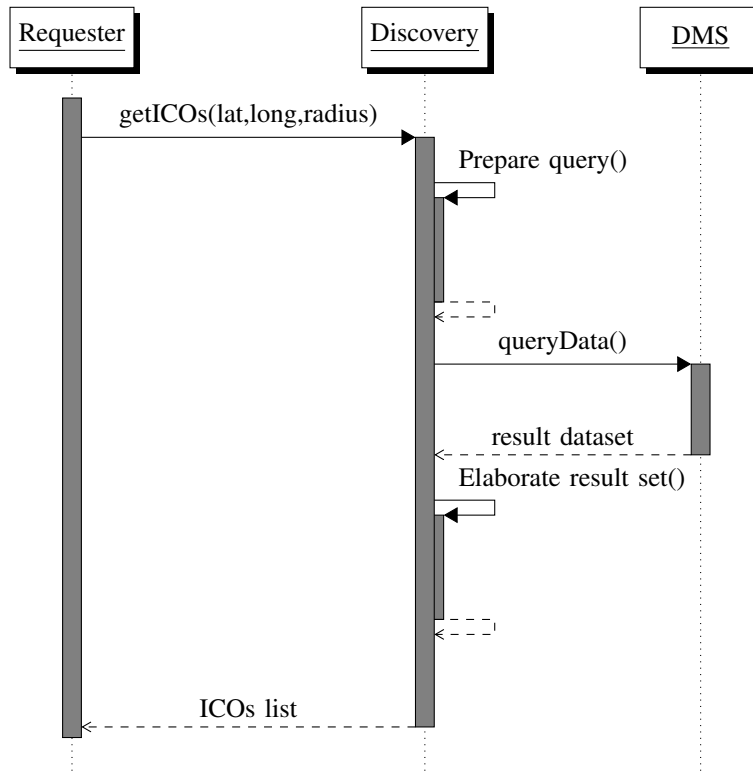


Fig. 2: SD - Example of interactions.

III. DISCOVERY OF “RELEVANT” RESOURCES

The Discoverer module - horizontally integrated in the platform itself as shown in Figure 1 - is responsible for discovering ICOs, Systems, and Services inside VITAL. Although the discovery of Systems and Services represents an important business for the whole architecture, in this paper, we only focus on the ICOs.

The SD offers its functionality to other modules, via a RESTful web service [7]; queries are embedded in the body of an HTTP request and represented using JavaScript Object Notation (JSON) Standard. In the following, we describe the main interfaces exposed by the SD and their features.

The discovery process is performed on data stored in the DMS; this operation is achieved without regard to the underlying platform handling the ICOs. This represents, therefore, a key feature as the platform agnostic property is one of the goals proposed by the architecture. Internet-Connected Objects being part of VITAL, are described by a set of metadata that characterizes the nature of an element. Thanks to this information, ICOs can be classified according to their properties. Such a classification is essential for the later discovery of all the elements stored in the system and that suit the criteria defined by users. The main properties that can drive a discovery process are:

- **Position.** It represents the last known position of an ICO - expressed in latitude and longitude coordinates -. Such values can be immutable in the case of a *stationary node* or it can change over time for a *mobile node*.

- **Mobility.** It describes the mobility type of an Internet-Connected Object. Every node can be configured as being *Stationary*, *Mobile*, or *Predicted* - in this latter case the mobility pattern is known -.
- **Connectivity.** It provides information about the capability of a node to be connected over the Internet and a description of the connection stability.
- **Observation capability.** It describes all the properties that a node can observe (e.g. Speed, Footfall).

All the above parameters are modelled according to the VITAL ontology; a detailed list of those properties is available in Table I.

Listing 1: SD - Example of request.

```

{
  "position": {
    "radius": 10,
    "longitude": 3.147887,
    "latitude": 50.605792
  },
  "observes": "vital:Speed",
  "type": "vital:VitalSensor"
}

```

Figure 2 shows how the SD interacts with other modules in order to execute an assigned task. Listing 1 describes an example of a request in JSON; the target, in this case, is the discovery of ICOs that can observe *Speed*, in the Area of Interest defined by the *position* properties.

TABLE I: SD - ICOs request parameters.

NAME	TYPE	DESCRIPTION
<i>type</i>	String	Category assigned to ICO
<i>position</i>	Object	Keys to perform discovery over spatial region
<i>latitude</i>	Number	Latitude's value expressed in WGS84 standard
<i>longitude</i>	Number	Longitude's value expressed in WGS84 standard
<i>radius</i>	Number	Distance in Kilometers from the center or the Area of Interest
<i>observes</i>	String	Measuring properties provided by an ICO
<i>movementPattern</i>	String	Select ICO registered with a specific movement pattern
<i>connectionStability</i>	String	Select a specific level of connection stability
<i>hasLocalizer</i>	Boolean	Select ICO with localizer services (i.e., GPS)
<i>timeWindow</i>	Number	Time in minutes; represents the time window for position estimation

Upon reception of a request, the DISCOVERY re-formulates the query. In order to perform a geographical search, it is necessary to compute latitude and longitude boundaries. Given a centre position (*lat - long*) and a distance (*radius*) in kilometers, the conversion process aims to compute minimum and maximum values for both latitude and longitude. This computation is performed using the law of haversine [8] defined in spherical trigonometry, which relates sides and angles of spherical triangles. Once the query is ready and modelled according to the VITAL ontology, it is sent to the DMS, which will respond with a data-set of available ICOs matching the request parameters. At this point, the SD applies its rules - giving priority to ICOs that are closer to the Point of Interest, have good connection, etc. - in order to provide back to the REQUESTER an organized list of ICOs.

In order to have a time estimation about the above operations, we deployed the Discoverer on WildFly² - an application server written in Java, which implements JAVA EE specifications -. It is executed on a machine with 8 GB of RAM and a processor Intel Core i5 (2.6 GHz). Figure 3 resumes the main results; we can observe that the SD takes, in average, 2 seconds in order to discover a generic ICO, System, or Service registered in the VITAL's DMS. To avoid time out issues, we implemented a timer of 5 seconds, after which the SD returns with an error to the REQUESTER. Regarding the CPU, each query is performed in 32 milliseconds.

IV. RELATED WORK

One of the most important challenges within the IoT context, is the discovery of appropriate data-sources that satisfy

²<http://wildfly.org>

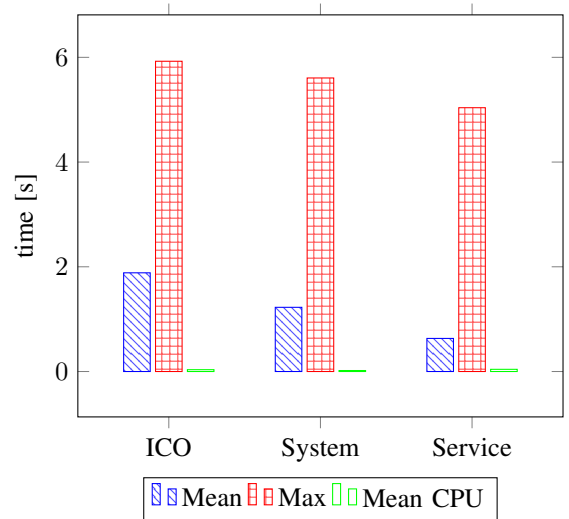


Fig. 3: SD - Performance.

user requirements. In order to deal with this problematic, many solutions have been proposed in literature.

The discovery is an operation that can be performed directly *in-network* (i.e., the detection of neighbors) and/or *out-network* (i.e., a middleware that aims to discover resources that belong to different networks). The first case is something more related with the building and maintenance of the network itself, especially in order to introduce new services and devices; a variety of protocols exists and is used nowadays for TCP/IP networks [9]; to name the most well known, Service Location Protocol (SLP), developed by IETF and Universal Plug and

Play (UPnP) developed by Microsoft. A lot of effort has also been done for resource-constraint networks; in [10], Shelby et al. present COAP; in this case, services are published into a register - which can be queried by users - stored at the gateway. In order to overcome the limitations of a centralized solution, in [11] we proposed a ranking mechanism for Sensor Networks that facilitates the discovery of services; thanks to our proposal indeed, sensor nodes can discover the neighborhood and, at the same time, the services offered just by adding few bytes of overhead (10 bytes for each service) to their discovery protocol.

When we consider *out-network* discovery, the challenge concerns more the representation of different data-sources and how users can easily access those resources and the provided services. In this context, many middleware solutions have been proposed; in the following, we present the most representative.

Global Sensor Networks (GSN) ³ is a middleware that aims to address the challenges of sensor data integration and distributed query processing. The main purpose is to make applications hardware-independent. GSN lacks semantics to model meta-data.

OpenIoT ⁴ concentrates on providing a cloud-based middleware infrastructure in order to deliver on-demand access to IoT services, which could be formulated over multiple platforms. Anyway, it is not designed to deal with Smart City scenarios.

Sensor-Cloud [12] aims at managing physical sensors by connecting them to the cloud; sensor devices should be described by using SensorML. It does not take the management of entire systems and/or services into account.

From the above solutions, it is evident that using semantic technologies is recognized as a good enabler in order to describe resources across different IoT platforms and data-sources. Once modelled, the users should easily access to those resources, select the one that most suits his/her specific business context. In some circumstances, indeed, the user is just interested about a generic ICO that can observe, in an Area of Interest, a defined property (e.g., Speed and/or Temperature); therefore it is important to provide the “best” candidate resource. Among other functionality, this is what our Discovery module achieves.

V. CONCLUSION

In this paper, we presented the VITAL architecture and how it deals with the fragmented landscape that nowadays exists among multiple IoT platforms and ecosystems. We contextualized VITAL within the Smart City concept, and we described in detail its ICOs and Services Discovery module.

The discovery of appropriate data-sources represents, indeed, an important task, specially in a context where a huge number of resources is available (i.e., a Smart City). The SD within VITAL aims to address this problematic by taking account of different parameters in order to evaluate and better organize the ICOs that best suit the business context defined

by users. Among those parameters, the classification in terms of mobility and connection stability plays a key role; indeed the Discovery returns, to the Requester, ICOs that have good connection to the Internet and that are available in the Area of Interest. Furthermore, this module represents a key feature in order to reach the platform agnostic property. VITAL indeed, aims to appear completely transparent for users in terms of standards and protocols used by the underlying connected platforms.

In the future, our intent is to improve our SD in terms of performance and accuracy of the results. To reach these objectives, the idea is to deal with scalability and to introduce and study new parameters (e.g., historical behavior) in order to better characterize each ICO.

ACKNOWLEDGMENTS

This work is partially supported by CPER Nord-Pas-de-Calais/FEDER DATA and by the European Community in the framework of the VITAL (Virtualized programmable InTerFAces for innovative cost-effective IoT depLoYments in smart cities) FP7 project under contract number FP7-SMARTCITIES-608662. The authors acknowledge help and contributions from all partners of the project.

REFERENCES

- [1] ITU Internet Reports, “The Internet of Things,” 2005.
- [2] R. Petrolo, V. Loscri, and N. Mitton, “Cyber-Physical Objects as key elements for a Smart Cyber-City,” in *Management of Cyber Physical Objects in the Future Internet of Things*, Feb. 2016.
- [3] Fast Co. Design, “By 2050, 70% Of The World’s Population Will Be Urban. Is That A Good Thing?” <http://www.fastcodesign.com/1669244/by-2050-70-of-the-worlds-population-will-be-urban-is-that-a-good-thing>, 2012.
- [4] R. Petrolo, V. Loscri, and N. Mitton, “Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms,” *Transactions on Emerging Telecommunications Technologies*, pp. 1–11, 2015.
- [5] M. Thoma, T. Braun, C. Magerkurth, and A.-F. Antonescu, “Managing things and services with semantics: A survey,” in *Proceedings of NOMS - IEEE Network Operations and Management Symposium*, Krakow, Poland, May 2014.
- [6] A. Roukounaki, J. Soldatos, R. Petrolo, V. Loscri, N. Mitton, and M. Serrano, “Visual Development Environment for Semantically Interoperable Smart Cities Applications,” in *Proceedings of EAI International Conference on Interoperability in IoT*, Rome, Italy, Oct. 2015.
- [7] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.
- [8] “Haversine formula,” https://en.wikipedia.org/wiki/Haversine_formula, last visited 27 February 2016.
- [9] C. Bettstetter and C. Renner, “A comparison of service discovery protocols and implementation of the service location protocol,” in *Proceedings of the 6th EUNICE Open European Summer School: Innovative Internet Applications*, 2000.
- [10] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” <http://www.rfc-editor.org/rfc/rfc7252.txt>, RFC Editor, Tech. Rep., 2014.
- [11] R. Petrolo, V. Loscri, and N. Mitton, “Confident-based Adaptable Connected objects discovery to HArmonize smart City Applications,” in *Proceedings of WD - IFIP Wireless Days*, Toulouse, France, Mar. 2016.
- [12] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure physical sensor management with virtualized sensors on cloud computing,” in *Proceedings of NBiS - 13th International Conference on Network-Based Information Systems*, Gwangju, Korea, Sep. 2010.

³<https://github.com/LSIR/gsn>

⁴<http://www.openiot.eu>