Integrating Wireless Sensor Networks within a City Cloud

Riccardo Petrolo*, Nathalie Mitton*, John Soldatos†, Manfred Hauswirth‡, Gregor Schiele‡
* Inria Lille-Nord Europe, France. e-mail: firstname.lastname@inria.fr
† Athens Information Technology, Greece. e-mail: jsol@ait.gr
‡ Digital Enterprise Research Institute, Ireland. e-mail: firstname.lastname@deri.org

***Abstract*—Smart City solutions are currently based on multiple architectures, standards and platforms, which have led to a highly fragmented landscape. In order to allow cities to share data across systems and coordinate processes across domains, it is essential to break these silos. A way to achieve the purpose is sensor virtualization, discovery and data restitution. In this paper, a federation of FIT IoT-LAB within OpenIoT is presented. OpenIoT is a middleware that enables the collection of data streams from multiple heterogeneous geographically dispersed data sources, as well as their semantic unification and streaming with a cloud infrastructure. Future Internet of Things IoT-LAB (FIT IoT-LAB) provides a very large scale infrastructure facility suitable for testing small wireless sensor devices and heterogeneous communicating objects. The integration proposed represents a way to reduce the gap existing in the Internet of Things (IoT) fragmentation, and, moreover, allows users to develop smart city applications by interacting directly with sensors at different layers. We illustrate it trough a basic temperature monitoring application to show its efficiency.**

***Keywords*—Smart Cities, IoT, FIT IoT-LAB, OpenIoT.**

I. Introduction

A Smart City is an urban system that uses Information and Communication Technology (ICT) to make both its infrastructure and its public services more interactive, more accessible and more efficient. Focusing on the technology needed to build smart cities, the development is not restricted to provide services independently and individually, but it will be necessary to deploy a whole infrastructure for city data collection, and transmission, storage and analysis of data for the supply of services to citizens [1]. In a smart city, Internet of Things (IoT) and especially sensor networks play a crucial role. Their data streams can be converted into useful information for infrastructure monitoring and management, smart street lightning, air quality monitoring, smart traffic regulation, etc.

IoT applications are currently based on multiple architectures, standards and platforms, which have led to a highly fragmented IoT landscape. This fragmentation is evident in the area of smart cities, which typically comprise several technological silos (i.e. IoT systems that have been developed and deployed independently). Nowadays there is a pressing need to remove these silos in order to allow cities to share data across systems and coordinate processes across domains, thereby essentially improving sustainability and quality of life. A way to achieve efficient systems is sensor virtualization, discovery and data restitution.

978-1-4799-5674-6/14/$31.00 © 2014 IEEE

To this purpose, in this paper, we propose the integration of nodes from an open experimental platform (FIT IoT-LAB) within OpenIoT. OpenIoT is an open source middleware that enables the collection of data streams from multiple heterogeneous geographically dispersed data sources, as well as their semantic unification and streaming with a cloud infrastructure for various later uses, allowing silo breaking.

Future Internet of Things IoT-LAB (FIT IoT-LAB) provides a very large scale infrastructure facility suitable for testing small wireless sensor devices and heterogeneous communicating objects over large scale. The FIT IoT-LAB testbed is composed of more than 2000 nodes over 8 sites across France [2].

The federation of the two platforms aforementioned, represents a way to reduce the gap existing in the IoT fragmentation. Moreover, the integration presented, allows users to develop smart city applications by interacting with sensors at different layers. FIT IoT-LAB, indeed, allow users to deploy their experiment without any kind of restrictions on the programming language, on the programming model or on the OS that he/she desires to use. Then, the virtualization of the data produced under an experiment, allowing users to define complex IoT applications, combining their data with data from other platforms available in the OpenIoT cloud. We illustrate it trough a basic temperature monitoring application to show its efficiency.

The remainder of the paper is organized as follows. In Section II we briefly review the literature and provide information about the semantically annotation of sensors data. Section III presents the OpenIoT middleware. Section IV describes the FIT IoT-LAB sensors. The details of our contribution, the platform integration are reported in Section V. Section VI concludes the paper.

II. Background and Motivation

In the previous section we introduced both IoT and Sensor Network (SN). In [3] authors explain the relationship between the two concepts, sensor networks can exist without the IoT. However, the IoT cannot exist without SN, because SN provides the majority of hardware (e.g. sensing and communicating) infrastructure support, providing access to sensors and actuators. The integration between wireless sensor networks and the IoT is totally discussed in [4]. However, their solution just focuses on the network, without considering the description of the data produced by SN. In [3] the use of semantic in IoT is recognized as one of the most important

functionalities to connect objects together. Barnaghi et al. [5] outlined the need for semantic annotation of IoT resources and services. This justifies our decision to annotate the sensors with related context information within an IoT middleware. In [6], authors explain the advantages of using cloud computing in order to enable IoT applications and introduce a middleware framework that aspires to enable the dynamic, for self-organizing formulation of optimized cloud environments for IoT applications.

The federation proposed in this paper considers the benefits of the semantically annotation of sensors data in the cloud, in order to create IoT applications using multiple heterogeneous geographically dispersed data sources. Moreover, thanks to the FIT IoT-LAB, the developer can also interacts directly with the sensors, without any kind of restrictions on the programming language, programming model or operating systems that he/she desires to use.

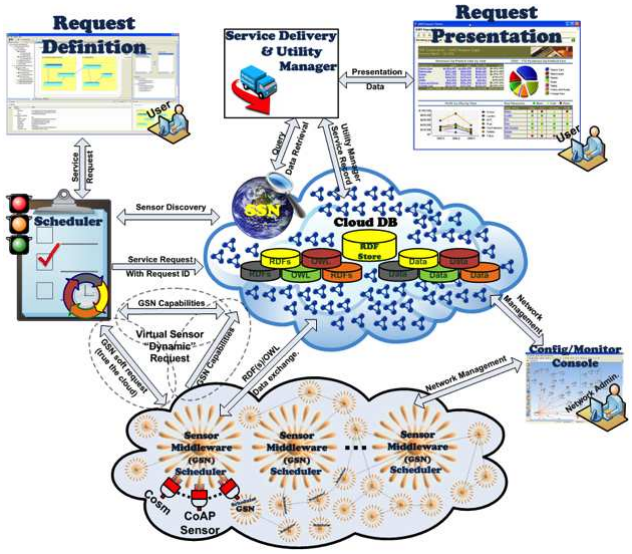


Fig. 1: Overview of the main modules of the OpenIoT open source middleware platform.

III. OpenIoT Cloud Middleware Infrastructure

In order to leverage the benefits of cloud computing infrastructures for streaming, aggregating and processing WSN data, we make use of an open source middleware platform for integrated IoT applications in the cloud [6]. In particular, we integrated WSN data from the FIT IoT-LAB[1] experimental platform to the OpenIoT middleware platform[2], which enables the collection of data streams from multiple heterogeneous geographically dispersed data sources, as well as their semantic unification and streaming with a cloud infrastructure [7]. The OpenIoT middleware infrastructures comprises the following main modules (illustrated in Figure 1):

- **Global Sensor Networks (GSN) middleware.** The GSN middleware[3] is a popular open source sensor middleware [8], which enables the collection of data streams from virtually any sensors. GSN defines different sensors and data sources on the basis of a basic set of metadata, which comprise the description of a virtual sensor. In the scope of OpenIoT, the original GSN implementation has been enhanced with the capabilities of annotating sensor data and metadata according to the W3C SSN (Semantic Sensor Network) ontology [9]. Therefore in the scope of OpenIoT the GSN sensor middleware serve as basis for semantically unifying diverse sensor streams, which are collected/accessed based on the sensor middleware.
- **Directory Service.** The directory service is provided in the form of an RDF Store (annotated database using RDF format) also called triple store. This RDF Store provides a range of services for dynamically discovering sensors (e.g., based on their type and location), as well as for querying their data. Note also that OpenIoT deploys this RDF Store within a cloud environment (such as a private or public cloud), thereby realizing the IoT/cloud integration infrastructure.
- **Scheduler.** The Scheduler is the module that receives end-users requests for IoT services and accordingly discovers the requested sensors (based on the services of the directory). The scheduler undertakes also to reserve resources associated with the requested IoT service.
- **Service Delivery & Utility Manager (SD&UM).** The SD&UM combines the various data streams as indicated/required by the IoT services that are communicated to the Scheduler. To this end, this component makes use of the service description and resources identified and reserved by the Scheduler component. On the other hand, this component acts as a service metering facility which keeps track of utility metrics for each individual service. Metering can be used to drive functionalities such as accounting, billing and utility-driven resource optimization, which are are essential in the scope of a utility (pay-as-you-go) computing paradigm.
- **Request Definition UI.** This is a visual tool supporting the discovery of sensor models, as well as the development (in a graphical fashion) of IoT services. This module interacts with is the scheduler in order to communicate requests for IoT services (as the latter are communicated by the various users).
- **Request Presentation UI.** A basic functionality of requesting of available services of a specific user, polling for data regarding a specific service and visualizing them is supported by this module. To this end, the module interacts with the SD&UM. In order to integrate physical and/or virtual sensors to the cloud, one has to interface them to the GSN middleware, which provides a versatile way for defining virtual sensors and deploying them with the middleware based on minimal programming effort. Specifically, the integration of a Wireless Sensor Network (WSN) to the cloud entails the description of the WSN as a GSN Virtual Sensor [8] and the subsequent im-

[1] http://fit-equipex.fr
[2] https://github.com/OpenIotOrg/openiot/
[3] https://github.com/LSIR/gsn

plementation of a driver for acquiring/managing data from the underlying sensor. GSN will undertake the announcement of the sensor to the cloud, as well as the streaming of the WSN data to the OpenIoT cloud.

IV. FIT IoT-LAB

The FIT IoT-LAB hardware infrastructure [2] consists of three main components: *(i)* the open wireless sensor node dedicated to the user; *(ii)* the full **FIT IoT-LAB node** that encompasses the open node also includes a gateway and a closed wireless node; *(iii)* the global networking backbone that provides power and connectivity to all FIT IoT-LAB nodes and guarantees the out of band signal network needed for command purposes and monitoring feedback.

The open wireless sensor node is made available to the user during his/her experimentation. This node is totally open and the user is granted a full access to the memory. The nodes are based on a low power MSP430-based platform and equipped with Texas Instruments CC1101 radio interfaces or CC2420, depending on the site.

The full FIT IoT-LAB node includes a control node, which has the specific role to control the open one, and also a gateway board to insure the control and management of the platform. The control node can be configured to poll temperature, light, acoustic activity and signal features such as RSSI.

To enable users to interact with the nodes of their running experiment, FIT IoT-LAB provides a complete Linux environment virtualization.

V. FIT IoT-LAB in the OpenIoT Cloud

FIT IoT-LAB, as aforementioned, provides appropriate tools, methods, experimental facilities for testing and managing large scale wireless network experiments. The platform is generic, open and flexible: it means that a user is able to remotely access and deploy his/her test without any kind of restrictions on the programming language, on the programming model or on the OS that he/she desires to use.

Integrating the data produced during a FIT IoT-LAB experiment in the OpenIoT cloud, represents a way to reduce the gap existing in the IoT fragmentation.

Once data is stored and semantically annotated on OpenIoT, its tools allow the representation of an IoT application through a graph, which can be modelled visually. A service graph describing a particular application can be created by dragging nodes from the node-toolbox into the application workspace, setting them up and wiring them together. Specifically, a service graph visualizes the information flow from a source (typically a sensor type) to a sink (a visualization widget). Therefore, all service graphs should contain at least one source node and one sink node. Other node types can be injected between the source and the sink to manipulate the data (i.e. perform an aggregation or filtering function). The tool can automatically generate the SPARQL code corresponding to the modelled graph(s). Accordingly, it can ensure the deployment of the graph based application over the OpenIoT middleware infrastructure, which is described in Section III.

In order to test the reliability of the proposed integration we decided to create an application, which shows the average of the temperature of Grenoble and Lille FIT IoT-LAB sites.

A. FIT IoT-LAB wrappers to GSN/OpenIoT

For the need of the application, a node from Grenoble and a node from Lille, have been reserved through the FIT IoT-LAB web portal, and upon deployment of the firmware that allows the nodes to poll TEMPERATURE, the test is launched. In the meanwhile, a python script, stored in the user virtual machine, will collect data from serial link, format them and create a csv file which contains as entry the sensing values and the correspondent data date.

For security/privacy requirements, it is not allowed to access resources from outside the FIT IoT-LAB platform, therefore, the csv file will be uploaded via FTP *(File Transfer Protocol)*, on a server running OpenIoT.

FIT data needs to be processed using a so-called wrapper, included in the GSN module, in order to be annotated semantically and then stored in the cloud. A wrapper is a piece of Java code that acquires data from virtually any sensor, including physical devices, sensor processing algorithms, social media processing algorithms and more. To support a rapid deployment, the virtual sensors are provided in human readable declarative forms (XML).

Listing 1: FIT IoT-LAB virtual sensor definition.

```
1 <virtual-sensor name="fitlille" priority="10" publish-to-lsm="true">
2     <processing-class>
3         <class-name>org.openiot.gsn.vsensor.LSMExporter</class-name>
4         <init-params>
5             <param name="allow-nulls">false</param>
6             <param name="debug-mode">false</param>
7         </init-params>
8         <output-structure>
9             <field name="temperature" type="double" description="descr"/>
10        </output-structure>
11    </processing-class>
12    <description>fit lille station</description>
13    <life-cycle pool-size="10"/>
14    <addressing>
15        <predicate key="geographical">FIT IoT-LAB - Lille</predicate>
16        <predicate key="LATITUDE">50.633579</predicate>
17        <predicate key="LONGITUDE">3.020366</predicate>
18    </addressing>
19    <streams>
20        <stream name="input1">
21            <source alias="source1" sampling-rate="1" storage-size="1">
22                <address wrapper="csv">
23                    <predicate key="file">data/fit_lille.csv</predicate>
24                    <predicate key="fields">timed, temp, visib</predicate>
25                    <predicate key="formats">timestamp(d/M/y H:m), numeric, numeric</predicate>
26                    <predicate key="bad-values">NaN,6999,-6999,null</predicate>
27                    <predicate key="timezone">Etc/GMT+1</predicate>
28                    <predicate key="sampling">5000</predicate>
29                    <predicate key="check-point-directory">csv-check-points</predicate>
30                </address>
31                <query>select * from wrapper
32                </query>
33            </source>
34            <query>select timed, temp as temperature from source1</query>
35        </stream>
36    </streams>
37 </virtual-sensor>
```

Listing 1 shows the FIT IoT-LAB virtual sensor definition. The unique name (line 1) and the addressing (lines 14-18) keys are used to localize and to refer the sensor. Lines 8-10 show the structure of the output data stream, whereas the structure of the input streams is learnt from the specifications in lines 20-36. The attribute **wrapper="csv"** (line 22) indicates that the data stream is obtained through a csv file, which contains data. The format of this information is defined in line 25, line 26 specifies some bad-values that are not admitted. The *storage-size* defines the window size of 1 stream element. An SQL query specifies the processing of the source, by referring to the wrapper.

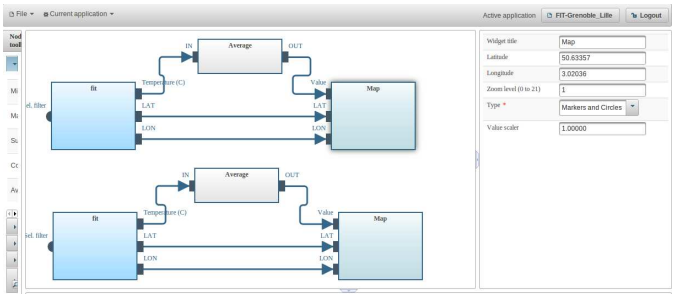
The definition aforementioned allows data from our experiment, run on FIT IoT-LAB, to be semantically annotated according to the W3C Semantic Sensor Network (SSN), available in the cloud, and accessible and processable from OpenIoT.

B. Using FIT IoT-LAB (semantically annotated) sensors

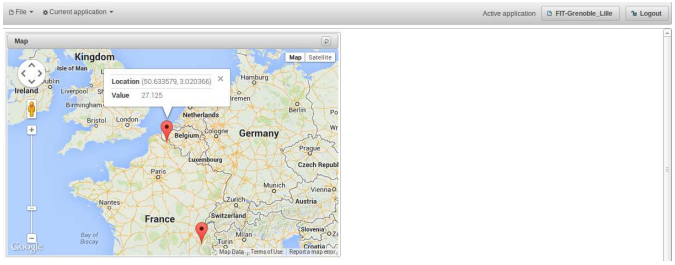
Following the streaming of FIT IoT-LAB data within the OpenIoT cloud infrastructure, the OpenIoT tools can be used to facilitate application development.

The Request Definition module is used to discover FIT IoT-LAB data through the "sensor discovery dialog". Since the application should contains at least one sink node, we decide to utilize a map, where the node's inputs are displayed using markers or circle overlays. Figure 2(a) shows the definition of the application, where, in order to localize source nodes, their LATITUDE and LONGITUDE are connected to the map. The sink value input endpoint can only accept aggregated values, therefore an aggregation node (the average in our example) is required between the nodes and this value input.

The results of our application are now ready to be visible in the Request Presentation (Figure 2(b)), where, it is possible to see on Google Maps the sensors used in our FIT IoT-LAB experimentation and to reading the correspondent average temperature.



(a) Request Definition.



(b) Request Presentation.

Fig. 2: User interface components of the OpenIoT tools.

VI. CONCLUSION

In this paper, we have introduced the integration of FIT IoT-LAB and OpenIoT. The first platform allows users to test small wireless sensor devices and heterogeneous communicating objects, while the second one enables the collection of data streams from multiple heterogeneous geographically dispersed data sources, as well as their semantic unification and streaming with a cloud infrastructure.

The virtualization of the FIT IoT-LAB data experiment on the OpenIoT cloud represents an implementation to reduce the gap existing in the IoT fragmentation, and, moreover, allows users to develop smart city applications by interacting with sensors at different layers.

The tools seen in the paper are, indeed, appropriately enhanced in order to facilitate smart cities application development.

The OpenIoT infrastructure is an ideal middleware platform for smart cities applications. This is because it can aggregate (in the cloud) a wide range of data sources regardless of their source platform or application. Hence, it provides opportunities for integrating smart city data from various heterogeneous platforms, while at the same time ensuring their interoperable representation based on the W3C SSN semantic representation. The latter should be extended in order to handle entities and concepts associated with: smart city application information, city-wide information, smart city stakeholder information, IoT system information, and device information.

ACKNOWLEDGEMENT

This work is partially supported by CPER Nord-Pas-de-Calais/FEDER CIA and by the European Commission in the framework of the VITAL FP7 project.

REFERENCES

[1] S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara, and A. G. Skarmeta, "A Global Perspective of Smart Cities: A Survey," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, Jul. 2013, pp. 439–444.

[2] C. B. des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noël, and J. Vandaele, "Using SensLAB as a first class scientific tool for large scale wireless sensor network experiments," in *NETWORKING 2011*. Valencia, Spain: Springer-Verlag, May 2011, vol. NETWORKING, pp. 147–159.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[4] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless sensor networks and the internet of things: Do we need a complete integration?" in *1st International Workshop on the Security of the Internet of Things (SecIoT'10)*, IEEE. Tokyo (Japan): IEEE, December 2010, p. xxxx.

[5] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things," *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012.

[6] J. Soldatos, M. Serrano, and M. Hauswirth, "Convergence of Utility Computing with the Internet-of-Things," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, Jul. 2012, pp. 874–879.

[7] M. Serrano, M. Hauswirth, N. Kefalakis, and J. Soldatos, "A Self-Organizing Architecture for Cloud by Means of Infrastructure Performance and Event Data," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1. IEEE, Dec. 2013, pp. 481–486.

[8] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks," in *2007 International Conference on Mobile Data Management*. Conference and Custom Publishing, May 2007, pp. 198–205.

[9] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.