GENERIC MODULE FOR COLLECTING DATA IN SMART CITIES

Alicia Martinez [a], Fernando Ramirez [a], Hugo Estrada [b], L.A. Torres [a]

[a] CENIDET, Department of Computation Sciences, 80626 Palmira, 62490 Cuernavaca, Mor., México - (amartinez, fernandoramirez16c, leontorres16c)@cenidet.edu.mx
[b] INFOTEC, Av. San Fernando 37, Toriello Guerra, 14050 Mexico City, México- hugo.estrada@infotec.mx

KEYWORDS: Smart Mobility, Internet of Things, Smart Cities

ABSTRACT:

The Future Internet brings new technologies to the common life of people, such as Internet of Things, Cloud Computing or Big Data. All this technologies have change the way people communicate and also the way the devices interact with the context, giving rise to new paradigms, as the case of smart cities. Currently, the mobile devices represent one of main sources of information for new applications that take into account the user context, such as apps for mobility, health, of security. Several platforms have been proposed that consider the development of Future Internet applications, however, no generic modules can be found that implement the collection of context data from smartphones. In this research work we present a generic module to collect data from different sensors of the mobile devices and also to send, in a standard manner, this data to the Open FIWARE Cloud to be stored or analyzed by software tools. The proposed module enables the human-as-a-sensor approach for FIWARE Platform.

1. INTRODUCTION

Nowadays, most of Future Internet technologies are related to Internet of Things (PPP, 2015), also incorporating new technologies as cloud computing or Big Data. Several development platforms have been developed to support IoT such as: Smart+Connected Communities (Cisco, 2010), Amazon Web Services IoT (Amazon, 2015), IBM Watson IoT Platform (Macgillivray, 2016), FIWARE (FIWARE-ABOUT US, 2016), etc. In the case of FIWARE Platform, it is compose by a set components, called Generic Enablers, which are components developed for horizontal purposes that could be used to implement robust applications in very different applications domains. However, the implementation of news applications in all mentioned platforms, including FIWARE, require specialized knowledge by developers, for example OpenStack (OpenStack, 2017) technologies, and also different development languages, for example, Angular to define the front-end of the applications.

FIWARE has been designed, from the beginning, to be an open and standard platform, based on open source to foster the creation of the needed standards to develop smart services and applications in different domains, such as smart cities, logistics, energy, agriculture, smart industry, etc. (Telefonica, 2017).

The standard proposed by FIWARE is designed to manage the entire life cycle of the context information, including updates, queries, registrations and subscriptions. This standard communication protocol is called FIWARE-NGSI (FIWARE-NGSI-V2, 2017a). This protocol permits developers to use different kind of sensors, speaking different protocols, and unify all the communication of data to be stored in the FIWARE Cloud. The main elements of the NGSI data models are contextual entities, attributes and metadata.

One on the main characteristics of Internet of Things applications is the use of smartphones as source of information of users. This applications use the resources of smartphones to produce data about the user context, this is the case of applications as Waze, Google Maps, Uber, etc. However, the review of FIWARE Literature and Generic Enabler catalogue demonstrate that there is not a generic component for mobile devices that enables the context information management based on the standard NGSI. This is a relevant lack because the increasing of the concept of human as a sensor based on mobile devices.

The European Project SmartSDK has the objective of developing data models and reference architectures for smart services based on FIWARE. One of the key components is the development of a generic component for the creation of mobile applications that obtain data from the smartphones and sent this data to the FIWARE Cloud based on the NGSI standard (FIWARE-NGSI-V2, 2017b). The specific Generic Enabler that manages the context data in the cloud according with NGSI is called Orion Context Broker.

2. BACKGROUND

2.1 Smart cities

A smart city is an urban development vision to integrate Information and Communication Technology (ICT) and Internet of Things (IoT) technology in a secure fashion to manage a city's assets. The smart city can be defined as a city that uses information and communication technologies to improve the critical infrastructure of the city and also to make more efficient the public services for citizens (Cisco, 2010).

A Smart City uses technology to improve the quality of life and the accessibility of its inhabitants. Additionally, it ensures sustainable economic, social and environmental development in permanent improvement. A smart city allows citizens to interact with it in a multidisciplinary way and also permits the city to adapt in real time to their needs (OpenStack, 2017).

2.2 FIWARE

FIWARE is an open source API platform, which, to date, has released 42 standardized software components aimed at helping startups and enterprise build the next generation of smart applications and services for cities, industries, e-health or agribusiness.

FIWARE is an open platform, which provides a set of tools for different functionalities. It is an innovation ecosystem for the creation of new applications and Internet services. It is especially useful in terms of Smart Cities, as it ensures the interoperability and the creation of standard data models.

The platform provides enhanced OpenStack-based Cloud capabilities and a set of tools and libraries known as Generic Enablers (GEs) with public and open-source specifications and interfaces. The use and management of data coming from "things" (i.e. sensors, actuators and other devices) is also a complex process, as there are many different protocols in the IoT sphere, but FIWARE provides a set of GEs allowing accessing the relevant information through only one API (NGSI). It not only allows to read the sensor information, but also to act on some elements. Therefore, Context Broker is an essential part of the architecture to collect data, analyze them on real time, consult archives and their analysis, as well as to publish them as open data from a city. On the other hand, other functionalities such as business intelligence, web interfaces and advanced interfaces allow the creation of very powerful applications and solutions.

2.3 Orion Context Broker

The Orion Context Broker is a key component of FIWARE to enable the data context ingestion and also to enable the subscription of applications to the data context. The main concept of the Context Broker is that data context producer can generate information and placed this in the cloud, without a previous knowledge of the users or application that will use the data. In this case, for example, for the case of smart mobility scenario, one user can generate a traffic alert that is sent to the Cloud, and later on, the rest of application users can receive the anonymized notification of the alert in their devices. In this context, the Orion Context Broker is a bridge that enable external applications to manage for IoT devices in a transparent and easy manner, hidden the complexity of managing the capture of the data (FIWARE-OpenSpecification, 2017).

Figure 1 shows the logical architecture of the Context Broker, with its main components and interactions with other actors (FIWARE-OpenSpecification, 2017).

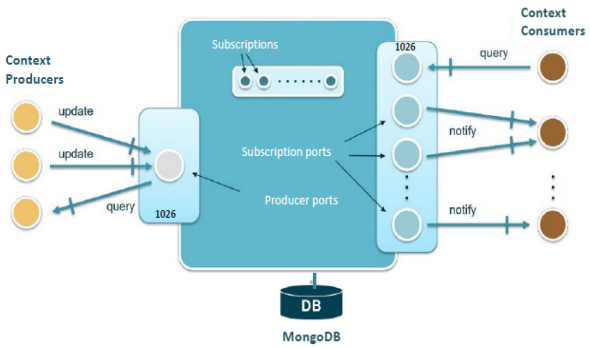

Figure 1 Logical architecture of the Context Broker.

Orion Context Broker allows developers to manage all the whole lifecycle of context information including updates, queries, registrations and subscriptions. The Orion Context Broker allows register context elements and manages them through updates and queries. In addition, the context information could be subscribed in order to receive some notification when the context element change (FIWARE-Orion, 2017).

The Publish/Subscribe service of the Context Broker GE supports two ways of communications: push, for the context produce, and pull for the context consumer (FIWARE-OpenSpecification, 2017). The Push communication occurs when the producer of data context sent data to the Context Broker and the Pull communication occurs when user or applications consume information stored in the Context Broker.

The Context information is represented through values assigned to attributes that characterize those entities relevant to your application. The Context Broker is able to handle context information at large scale by implementing standard REST APIs (FIWARE-Orion GE, 2016). The context information may come from many different sources for example: already existing systems, users, through mobile apps, sensor networks, etc. The Figure 2 shows the context broker scheme (FIWARE-Orion GE, 2016) where the Context Broker can receive context data for very different application domains.

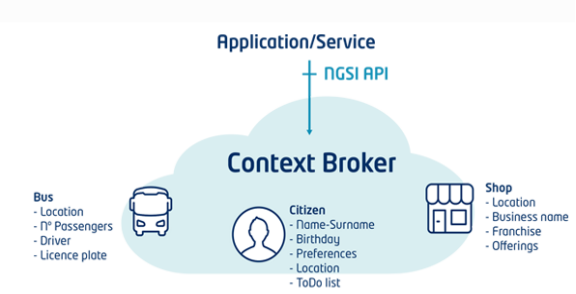

Figure 2 Context Broker Scheme.

One of the most important features of the Context Broker is that it allows to model and gain access to context information in a way that is independent from the source of that information (FIWARE-Orion GE, 2016).

2.4 Standard NGSI

The standard NGSI v2 of FIWARE is intended to manage the entire lifecycle of context information, including updates, queries, registrations, and subscriptions (FIWARE-NGSI v2-Specification, 2017).

The FIWARE NGSI (Next Generation Service Interface) API defines:

- A data model for context information, based on a simple information model using the notion of context entities.
- A context data interface for exchanging information by means of query, subscription, and update operations.
- A context availability interface for exchanging information on how to obtain context information.

The main elements in the NGSI data model are context entities, attributes and metadata, as shown in

Figure 3. The difference among these elements is that attributes describe the entity and metadata describe attributes.

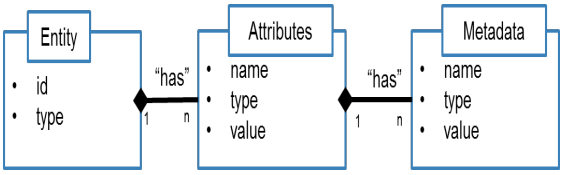

Figure 3 Elements of the NGSI data model.

Context entities, or simply entities, are the center of gravity in the FIWARE NGSI information model. An entity represents a thing, i.e., any physical or logical object (e.g., a sensor, a person, a room, an issue in a ticketing system, etc.). Each entity has an entity id. Furthermore, the type system of FIWARE NGSI enables entities to have an entity type. Entity types are semantic types; they are intended to describe the type of thing represented by the entity. Each entity is uniquely identified by the combination of its id and type.

Context attributes are properties of context entities. In the NGSI data model, attributes have an attribute name, an attribute type, an attribute value and metadata.

- The attribute name describes what kind of property the attribute value represents of the entity, for example current speed.
- The attribute type represents the NGSI value type of the attribute value. Note that FIWARE NGSI has its own type system for attribute values, so NGSI value types are not the same as JSON types.
- The attribute value finally contains: the actual data, optional metadata describing properties of the attribute value like e.g. accuracy, provider, or a timestamp.

Context metadata are used in FIWARE NGSI in several places, one of them being an optional part of the attribute value. Similar to attributes, each piece of metadata has:

- A metadata name, describing the role of the metadata in the place where it occurs; for example, the metadata name accuracy indicates that the metadata value describes how accurate a given attribute value is.
- A metadata type, describing the NGSI value type of the metadata value.
- A metadata value containing the actual metadata.

3. RELATED WORKS

The research work presented in (Chaovalit, Saiprasert, & Pholprasit, 2014) describes a method for driving event detection using SAX algorithm through a smarphone. This method detects the following events: braking, acceleration, left and right rotation, lane changes and U-turn. The sensors used for collecting data are: accelerometer, magnetometer and GPS (Global Positioning System) from a smartphone.

The SAX algorithm approximates a time series to discrete values; it is suitable for use in time series with noisy data. Also it has its own distance function based on the lower limit of Euclidean distance.

The research work (Cao, Lin, Zhang, & Zhang, 2017) proposes a procedure for automatically identifying driving events. Firstly, the data are collected from a three-axis accelerometer, and noise is eliminated applying three filters (Kalman filter, Wavelet filter and particle filter); Secondly, the Random Forest algorithm is used to classify the following events: starting, braking, left turn, right turn, left line shift, right line shift, U turn and right or left turn.

This research work (Martinez, Carlos Gonzalez, & Ricardo Carlos, 2014) proposes the automatic identification of interruptions or irregularities in the surface of the track, through the use of two algorithms: a neural network and a logistic regression method. Although the implementation of the neural network and logistic regression is done in different tools and languages, the data used in both tests are the same. The events detected in this research work are: bumpers, bumps, buoys, regular pavement and uneven pavement.

The research work (Gonzalez, Moreno, Escalante, Martinez, & Carlos, 2017) propose an algorithm for automatically detecting the following events: bumpers, asphalt stops, metal stops, irregular road and regular road. Their algorithm uses the BoW technique (Bag of Words) for extracting short segments of the signal coming from the accelerometer. The steps of the algorithm are: segmentation of all training measurements; each segment is represented by a descriptor; All descriptors corresponding to measurements of a class are grouped using the k-means algorithm in k-clusters.

SASVi (Martínez, González, & Carlos, 2014) is a system of assistance and road safety that provides resources and notify the driver of possible road irregularities. This platform was developed for the city of Chihuahua, which registers third place in road accidents nationwide. Speeding is the main cause of accident.

This platform uses the GPS and gyroscope of a smartphone to get the current position and orientation of the vehicle. The information from the GPS is processed to calculate the speed of the car and to compare the maximum speeds allowed. Also, the platform notifies the driver about possible encounters with bumps or speed reducers.

SenSafe (Liu, Wu, Zhu, & Zhang, 2016) is a framework for detecting automatically events of drivers and to alert it. The events detected are: left turn, right turn, left lane change, right lane change, left turn and right turn. The smartphone´s sensors used are: 3-axis accelerometer, magnetometer, gyroscope, and GPS.

This research work (Saiprasert, Pholprasit, & Thajchayapong, 2015) proposes the automatic detection of aggressive driving events; for to do this, three algorithms are implemented: Rules-based algorithm, Pattern matching algorithm and auto-trigger pattern matching algorithm.

The detection of the events uses the data coming from a smartphone. The driving events detected are: braking, abrupt braking, acceleration, abrupt acceleration, left turn, sharp left turn, right turn, sharp right turn, left lane change and lane change

4. GENERIC MODULE METHODOLOGY

This methodology, shown in Figure 4, has the objective to convert data coming from any mobile application into the NGSI standard from FIWARE. To perform this process, the data collected from the smartphones are converted using the module APK Smartphone. The data converted can be sent to the FIWARE Orion Context Broker. In this cases when an online connection can be used, the collected data are stores in a CSV in the smartphone. When an Internet connection is established, the data are sent to the FIWARE Context Broker using the NGSI standard. In case of private data this are sent to the Cygnus component to ensure the data privacy.

Following we present the details of the main components of the proposed methodology.

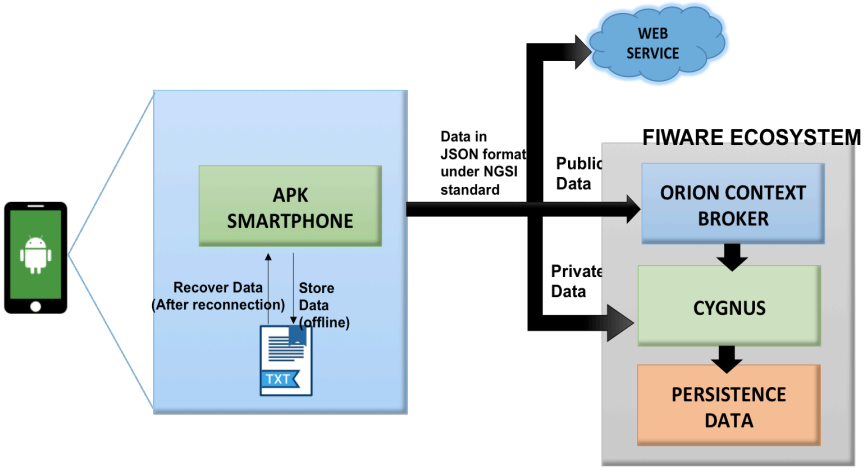

Figure 4 Proposed solution methodology

4.1 Module APK Smartphone

This module has the specific objective of transforming data collected by the smartphone in the NGSI standard communication protocol of FIWARE. The architecture is composes by 5 components (Figure 5).

- *Sensor Data:* This module obtains the data from the different smartphone ´sensors. The data quality will depend on the accuracy of the sensing devices in the phone. The module was developed in order to obtain the direct data of the smartphone and also to persist them in a storage.
- *Configuration:* This module configures all aspects related to the type of persistence of data. Some of the parameters considered are: Persistence method (local, WEB service, External Database or FIWARE Platform), connection chains and URLs, user data, etc.
- *Collection Data:* This module obtains the data of the sensors and integrates them with the data of the views. The data coming from the views correspond to the custom tags that the users can use to mark a specific event (for example, major adult drop, stop, point of interest, etc.) during data collection.
- *Business Rules and Entities:* This module receives the information collected from the mobile application in the original format. This module also translate this data in the standard NGSI.

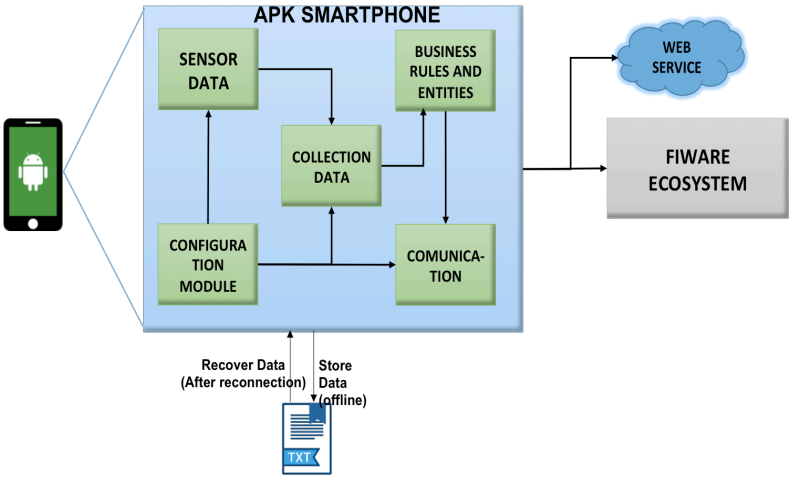

Figure 4.Architecture of our module proposed.

- *Communication:* This module obtains the data generated in the previous stage to be sent to one of the persistence modes considered in the module configuration (local, Web service, external database of FIWARE).

We have developed several storage schemas to cover the common scenarios that can be found in the use of the proposed component: a) if the user has an application, and wants to receive the data through a Web service, then the user must introduces the address of the Web service in module configuration; and, b) if the user wants to store the data in FIWARE ecosystem, then the user needs to connect to the Context Broker in case of public data, or the Cygnus component in case of private information.

4.2 JSON as interchange format

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language programing independent (JSON, 2017).

This module uses the JSON format as protocol for sending information. Some of the basic elements of standard JSON are the following:

- Object: an object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by: (colon) and the name/value pairs are separated by , (comma). Example:
  {
  "employee":{ "name":"John", "age":30, "city":"New York"}
  }
- Array: an array is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma). Example:
  {
  "employees":["John", "Anna", "Peter"]
  }

The NGSI standard of FIWARE incorporates some differences with the standard version. The additional information has been included to improve the richness of JSON format. The difference of JSON for NGSI are remarked in Table 1. For simplicity, the main difference is the use of id´s and type concepts.

It is important to point out that besides the representation of data in NGSI standard, the good practice of FIWARE imply the use of data models to represent the context information captured by the smartphones. The use of data models enable the applications to have standard manners to represent, store and communicate the context information. FIWARE has define several types of data models. In the case of the proposed module, the data models Device and Device model has been used to define the transfer of information to the FIWARE Cloud. In **¡Error! No se encuentra el origen de la referencia.**, an instance of these models is shown, in which a smart phone is used to capture the data from the accelerometer and the gyroscope. In this example, we use the data model Device that is composed by the entities: DeviceModel and Device. In accordance with FIWARE, a device is a tangible object that contains some logic and is producer and/or consumer of data. In table 1, three instances of the entity Device and *DeviceModel* are presented. *DeviceModel* is used to represent the invariant fields of the device. Device is used to represent the gyroscope and accelerometer. To identify the device type we use the category attribute, which in the example indicate one smartphone and two sensors.

The consisOf attribute is used to create reference to devices embedded in other device. This attribute is optional and it is used in the cases where an electronic devices is composed of several sensors. In the example, it is indicated that device smartphone is composed by the sensors accelerometer and the gyroscope.

Device (Smartphone)	DeviceModel

{
"id": "smartphone-345",
"type": "Device",
"category": "smartphone",
"osVersion": "Android 4.0",
"softwareVersion": "MA-Test 1.6",
"hardwareVersion": "GP-P9872",
"firmwareVersion": "SM-A310F",
"consistOf":[
"accelerometer-smartphone-345",
"gyroscope-smartphone-345"
],
"location": [18.876567, -99.63876],
"refDeviceModel": "deviceModel-smartphone-345",
"dateCreated": "2016-08-22T10:18:16Z"
}
	{
	"id":"deviceModel-smartphone-345",
	"type":"DeviceModel",
	"category":"smartphone",
	"brandName":"mySmartphoneBrand",
	"modelName":"S4Container 345",
	"manufacturerName":"mySensorInc.",
	"dateCreated":"2016-08-22T10:18:16Z",
	}

Device (Accelerometer)	Device (Gyroscope)

{
"id": "accelerometer-smartphone-345",
"type": "Device",
"category": "sensor",
"function": ["sensing"],
"controlledProperty": ["accelerometer"],
"hardwareVersion": "SMT-P9872",
"value": "-3.895|2.0493|9.87|2017-01-18T20:45:43.765Z-0800",
"dateCreated": "2016-08-22T10:20:16Z"
}
	{
	"id": "gyroscope-smartphone-345",
	"type": "Device",
	"category": "sensor",
	"function": ["sensing"],
	"controlledProperty": ["gyroscope"],
	"hardwareVersion": "SMT-P5672",
	"value": "45.895|7.0493|12.901|2017-01-18T20:45:43.765Z-0800",
	"dateCreated": "2016-08-22T10:20:16Z"
	}

Table 1. Example of the entities DeviceModel and Device based on the NGSI data model

4.3 Generic module implementation

A specific mobile application was developed to implement the proposed methodology. This application uses the following sensors: accelerometer, gyroscope, orientation and GPS.

Furthermore, the module was developed taking into account different environments and needs for data collection. The relevant aspects of the module are the following:

- *Persistence of data:* it is carried out of the following ways: a) using a remote database for which an internet connection is required; b) storage of data in the phone memory for which a flat file is used and where an internet connection is not required, c) use of a Web service; or to use of the FIWARE platform environment, for which an internet connection is required.
- *Geolocation:* The geolocation can be carried out using the GPS or the network, depending on the environment and the accessibility and connection characteristics.
- *Motion Sensors:* This module allows us to select the motion sensors, they are: accelerometer, gyroscope, orientation, and magnetometer. If some sensor is not included in the smartphone, its signal must be omitted.
- *Tagged Data:* All data obtained from the smartphone's sensors are tagged by default with a time stamp. Likewise, the module developed can to mark the data with the custom tags.

Figure 5, Figure 6, and Figure 7, show examples of windows of the module for data collection.

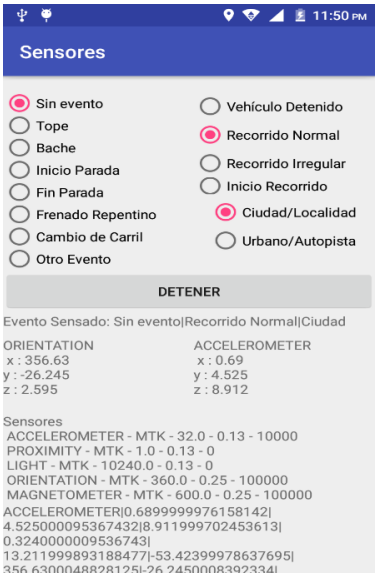

Figure 5 Functionality for tracking.

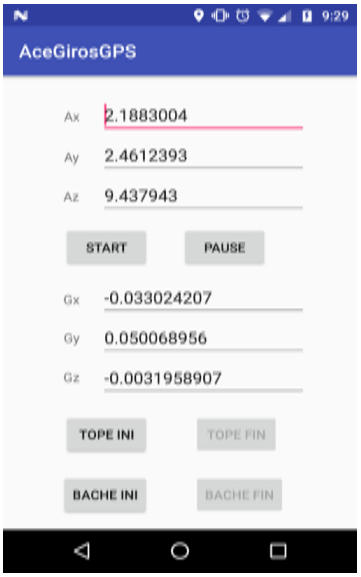

Figure 6 Functionality for marking stops and bumps.

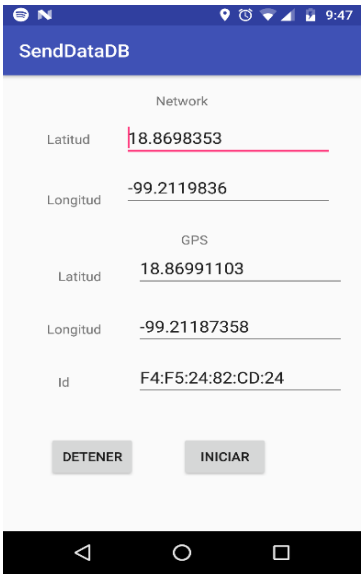

Figure 7 Functionality for georeferenced data.

5. EXPERIMENTATION IN APPLICATIONS

In this section we show several applications developed using the generic module. It is important to point out that developed module aims can be used in different domains. This is because the data obtained from the smartphone sensors can be used in any domain. For example, in the health area can be used to monitor aspects such as the physical activity of patients; in the transportation area, these can be used to monitor aggressive driver behavior or road irregularities. Likewise, the data obtained from the localization sensors are being used for route mapping and the geo-referencing of points of interest, etc.

In this section, we present two practical applications, which have used the proposed module for data collection. The first is a geolocation application, and the second is a routing application

5.1 Geolocation application

In this application, the generic module was used to determine if a smartphone is within a specific area. The data collected by the smartphone are sent to the FIWARE Context Broker. An application is subscribed to the Context Broker and asks it to forward all the information that comes from entities of type Device. The application allows to delimit areas under the standard geo-json (circumferences, polygons, squares, etc.) which are used to delimit areas. On the other hand, the NGSI API contains a set of functions that allows queries based on geographic coordinates. In this example, the module is used to take the location of the smartphone, each time a change is registered, the module sends the data to the Context Broker, and then the Context Broker sends the data to the subscribed application, which takes the data and the Draws on a map and reports if your smartphone is within a specific area.

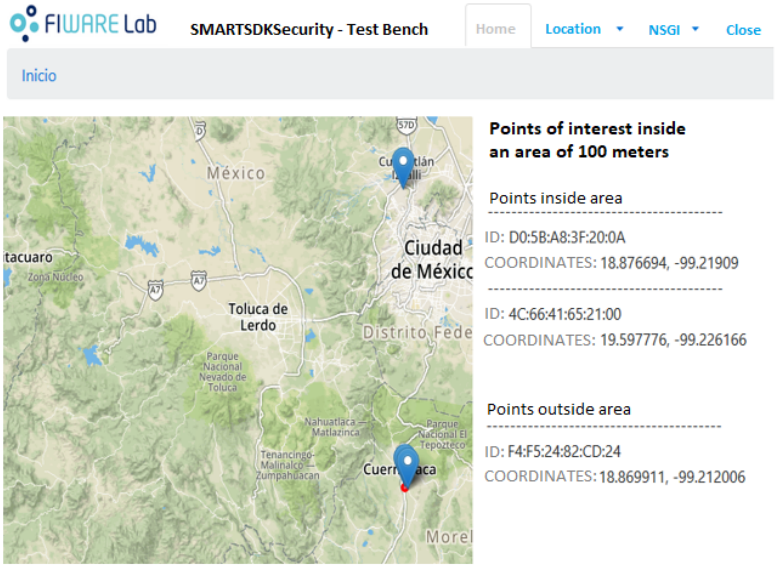

Figure 8 Example of a Geolocation application.

5.2 Tracking mapping application

In this application, we use our module to obtain the latitude and longitude data and to draw on a map the route registered by the smartphone. This application can be used to draw different paths. For example, the tracing of transport routes, tracking of parcels, etc. It can also be used to demarcate sites of interest and route stops.

The general operation of the application is as follows: The module collects the location data using the smartphone resources. This data is sent to an external database. Subsequently, the application connects to the database to obtain the history of the stored data. Once the application gets the history of the data, they are painted on a map. This way you can observe the route taken. Figure 9 shows an example of a tracking mapping application.

Figure 9 example of a tracking functionality

6. CONCLUSIONS

The module developed in this research permits developers connect mobile devices with FIWARE Orion Context Broker to collect information from the sensors of Android smartphones. This data are sent to the FIWARE Cloud and therefore, can be used by any application subscribed to that Context Broker. The data that are collected are the accelerometer, gyroscope, GPS, etc. The module was developed to be integrated in a clear and easy way to the FIWARE ecosystem. The proposed module can be used as data context producer for application of smart mobility, smart health or smart security, where the mobile device can be used as a sensors unit.

The module was developed taking into account connection and no-connection scenarios and also considering different means to the persistence of the collected data. We have three scenarios where the module can be used: a) the case where the user only needs to collect data and process it with external software tools. In that case, the only activity needed for user is the installation of the module in the smartphone and take the file that the module produces. b) Other scenario is the integration of the

module with a current system. In that case, the user only needs to install the module and indicate the specific data of the Web Service to be used. c) Other scenario is the users that wanted to store the information collected by the mobile devices in a database. It is important to point out that current version of module only considers the connection to MySQL databases.

As future work, we will incorporate the connection to other databases engines and also incorporate more sensors form the smartphone. One important challenge of this research is the creation a second version of module for several mobile operating systems. Other future work is the creation of filters to mitigate the data noise and to prepare the data for different classifiers for data mining tasks.

ACKNOWLEDGEMENTS

This research work has been partially funded by European Commission and CONACYT, through the SmartSDK project.

REFERENCES

Amazon., 2015. AWS IoT - Amazon Web Services. https://aws.amazon.com/es/iot/ (18 Aug. 2017).

Cao, W., Lin, X., Zhang, K., & Zhang, L., 2017. Poster Abstract : Analysis and Evaluation of Driving Behavior Recognition Based on a 3-axis Accelerometer Using a Random Forest Approach. *IPSN '17 Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks,* pp. 303–304. https://doi.org/10.1145/3055031.3055060

Chaovalit, P., Saiprasert, C., & Pholprasit, T., 2014. A method for driving event detection using SAX with resource usage exploration on smartphone platform. *EURASIP Journal on Wireless Communications and Networking*.

Cisco., 2010. Smart+Connected Communities: Changing a City, a Country, the World, 8. http://www.cisco.com/c/dam/en_us/solutions/industries/docs/scc/09CS2326_SCC_BrochureForWest_r3_112409.pdf (18 Aug. 2017).

FIWARE-ABOUT US., 2016. ABOUT US FIWARE. https://www.fiware.org/about-us/ (18 Aug. 2017).

FIWARE-NGSI-V2., 2017a. fiware-ngsiv2. http://fiware.github.io/specifications/ngsiv2/stable/ (18 Aug. 2017).

FIWARE-NGSI-V2., 2017b. FIWARE-NGSI v2-Specification. http://fiware.github.io/specifications/ngsiv2/stable/ (18 Aug. 2017).

FIWARE-OpenSpecification., 2017. FIWARE.OpenSpecification.Data.ContextBroker. https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.ContextBroker (18 Aug. 2017).

FIWARE-Orion., 2017. Fiware-Orion. https://fiware-orion.readthedocs.io/en/master/ (18 Aug. 2017).

FIWARE-Orion GE., 2016. Orion GE Context Broker. http://fiware-iot-stack.readthedocs.io/en/latest/context_broker/ (18 Aug. 2017).

Gonzalez, L. C., Moreno, R., Escalante, H. J., Martinez, F., & Carlos, M. R., 2017. Learning Roadway Surface Disruption Patterns Using the Bag of Words Representation. *IEEE Transactions on Intelligent Transportation Systems,* pp. 1–13. https://doi.org/10.1109/TITS.2017.2662483

JSON., 2017. JSON. http://www.json.org/json-es.html (18 Aug. 2017).

Liu, Z., Wu, M., Zhu, K., & Zhang, L., 2016. SenSafe: A smartphone-based traffic safety framework by sensing vehicle and pedestrian behaviors. *Mobile Information Systems,* 2016. https://doi.org/10.1155/2016/7967249

Macgillivray, C., 2016. The Platform of Platforms in the Internet of Things. https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=WWL12355USEN&attachment=WWL12355USEN.PDF (18 Aug. 2017).

Martinez, F., Carlos Gonzalez, L., & Ricardo Carlos, M., 2014. Identifying Roadway Surface Disruptions Based on Accelerometer Patterns. *IEEE Latin America Transactions*, 12(3), pp. 455–461. https://doi.org/10.1109/TLA.2014.6827873

Martínez, F., González, L. C., & Carlos, M. R., 2014. SASVi – Sistema de Asistencia y Seguridad Vial. *Research in Computing Science,* pp. 76, 51–60.

OpenStack., 2017. Software , What is OpenStack? https://www.openstack.org/software/ (18 Aug. 2017).

PPP, F. I., 2015. Led by industry, driven by users Addressing the challenge of Internet development in Europe. https://www.fi-ppp.eu/ (18 Aug. 2017).

Saiprasert, C., Pholprasit, T., & Thajchayapong, S., 2015. Detection of Driving Events using Sensory Data on Smartphone. *International Journal of Intelligent Transportation Systems Research,* pp. 1–12. https://doi.org/10.1007/s13177-015-0116-5

Telefonica., 2017. FIWARE, el estándar que necesita el IoT. https://iot.telefonica.com/blog/2016/09/es-fiware-estandar-iot (18 Aug. 2017).