Private Public Partnership Project (PPP)

Large-scale Integrated Project (IP)

D.10.8: Report on creation of FIWARE Regional Centers in Brazil

Project acronym: FIWARE
Project full title: Future Internet Core Platform
Contract No.: 285248
Strategic Objective: FI.ICT-2011.1.7 Technology foundation: Future Internet Core Platform
Project Document Number: ICT-2011-FI-285248-WP10-D.10.8
Project Document Date: 2015-02-23
Deliverable Type and Security: Public
Author: FIWARE Consortium
Contributors: FIWARE Consortium

1 Introduction

1.1 Executive Summary

The present document provides a report about the activities conducted in Brazil by USP and UFU in order to creation and support the FIWARE Regional Centers in Brazil.

The reasoning behind the FIWARE Regional Center is to have a FIWARE Lab infrastructure, a relationship with the local innovation ecosystem where the center is located and supporting actions that put together the stakeholders.

The document presents each deployed Regional center by describing the deployed infrastructure and the stakeholders of each local innovation ecosystem at the cities of São Paulo and Uberlândia where each Regional Center is located.

In each locality, different proof of concepts applications were created. The goal behind creating a proof of concept application was to highlight FIWARE platform focusing selected local challenges and to be used as an instrument for the evangelism and training of local stakeholders.

Several supporting actions related to evangelism, training and dissemination were conducted at São Paulo and Uberlândia in order to accomplish this task are detailed on this report.

The creation of the FIWARE Regional Centers in Brazil at the cities of São Paulo and Uberlândia was the first step towards the deployment of Future Internet based services and applications in Brazil that can take advantage of the FIWARE platform; the report explores the perspectives and point to next steps that will contribute to this aim.

1.2 About This Document

This document consolidates activities conducted in Brazil in order to create the FIWARE Regional Centers in Brazil. The document details the work by describing the activities and actions conducted at São Paulo and Uberlândia by USP and UFU respectively. Moreover, the document discuss the results, perspectives and presents next steps for FIWARE in Brazil.

1.3 Intended Audience

The audience of this document is very wide. It targets anyone interested in the innovation areas of the Future Internet. As such, it comprises the FIWARE project partners, the Use Case Projects, the FI-PPP Architecture Board and the EC.

Those outside the PPP interested on the technology foundation of the Future Internet services in different vertical sectors may find also helpful references and ideas. This therefore also targets decision makers within companies, who consider the adoption of FIWARE technologies, investors in companies targeting implementation of FIWARE technologies, market researchers and analysts trying to evaluate the potential of FIWARE technologies

1.4 Structure of this Document

Unlike most of the deliverables in FIWARE, this document is not part of the wiki and therefore it is not generated from wiki pages. It follows however, the same template as the wiki generated documents.

This document is organized as follows: Section 2 presents each FIWARE Regional Center deployed at São Paulo and at Uberlândia, their infrastructure and the local innovation. Section 3 details the proof-of-concept applications created at each Regional Center. Section 4 present different actions and activities regarding evangelism, training and dissemination. Section 5 presents some perspectives that are related to each Regional Center and its particular ecosystem and Section 6 contains some concluding remarks and future work.

1.5 Typographical Conventions

Starting with October 2012 the FIWARE project improved the quality and streamlined the submission process for deliverables, generated out of the public and private FIWARE wiki. The project is currently working on the migration of as many deliverables as possible towards the new system.

This document was edited on Microsoft Word following the common template that is shared with the semi-automatic scripts out of a MediaWiki system operated by the FIWARE consortium.

1.6 Keyword list

FIWARE, FIWARE Lab, Brazil Regional Centers, Deployment, Smart Campus, Open Data, Innovation Ecosystem, Smart Cities, FIWARE Based Applications.

1.7 Changes History

Release	Major changes description	Date	Editor
V0.1	First draft of deliverable structure	2014-12-20	UFU
V0.2	First version for delivery with USP and UFU contribution	2015-01-06	UFU & USP
V0.8	First final version for UFU and USP review	2015-01-22	UFU & USP
V1.0	First final version for delivery	2015-02-23	UFU & USP

1.8 Table of Contents

2 FIWARE Regional Centers

This Section details the activities in order to create the FIWARE Regional Centers in Brazil. The particular work conducted at São Paulo and Uberlândia are described. Each Regional Center contains a FIWARE Lab infrastructure and a relationship with the local innovation ecosystem.

2.1 Regional Center at São Paulo (USP)

In this section, we will describe the infrastructure that USP is deploying for the FIWARE node.

The node was deployed at the Campus Cidade Universitária Armando de Salles Oliveira, located at São Paulo, São Paulo State in Brazil. Figure 1 presents the campus deployment and the buildings where the related infrastructure was installed.

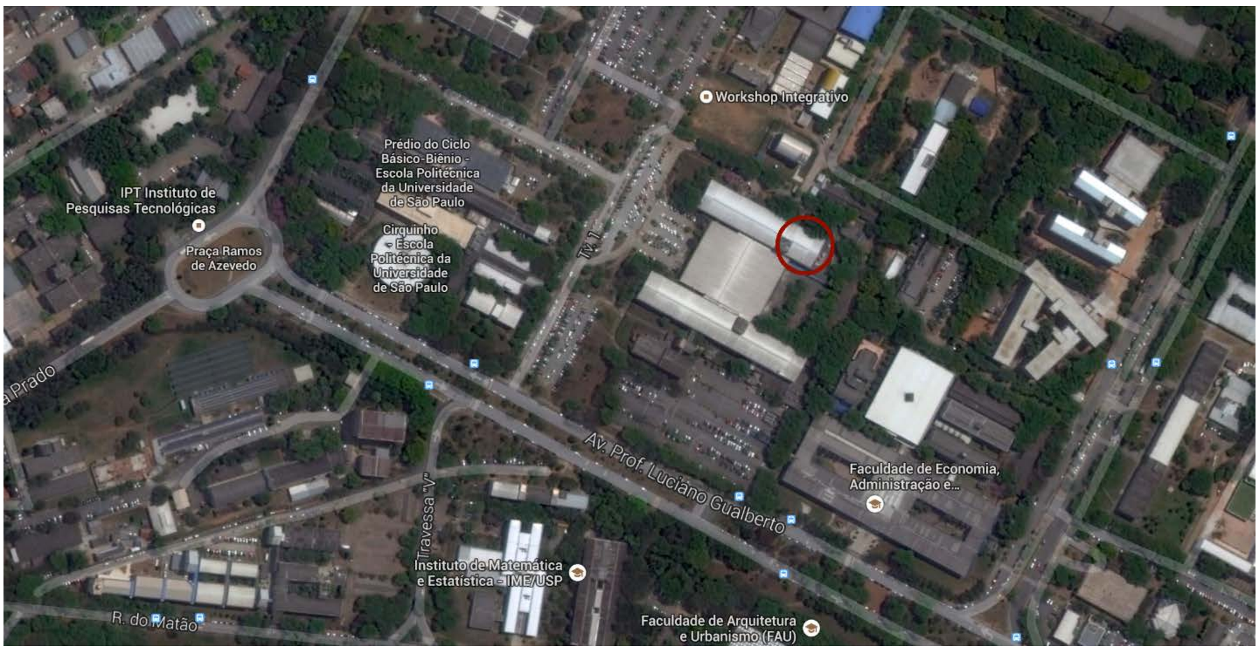


Figure 1 - Campus Deployment at São Paulo

2.1.1 Methodology

The deployment of the USP FIWARE node took place in three different stages:

- Testes in a virtualized environment
  The ITBox environment was installed over a virtualized infrastructure based on VirtualBox. The goal of this firs stage was to test and familiarize the USP team with the Fuel, Mirantis and OpenStack architecture. It was used to training purposes during the deployment period, and will be used in future training activities.
- Deployment on an off-the-shelf cluster

In this second stage, the deployment was done over a real computer cluster located in the High Performance and Pervasive Computing Systems Group Lab. The detailed hardware inventory can be found in Section 2.1.2.

- Final installation in a IBM eServer BladeCenter™

In this final stage, it was intended to provide a high-availability and high-performance infrastructure to the USP FIWARE Lab node. The detailed hardware description of the IMB eServer BladeCenter can be found in Section 2.1.2.

2.1.2 Computational Infrastructure

This subsection provides a detailed description about the selected hardware. The USP FIWARE Lab infrastructure has two separated Systems: a Private Cloud System, for USP internal users and intended to development and academic purposes, and another one for all FIWARE Lab users and intended for development of applications. The first one is a small computer cluster, with five dedicated servers and two switches. Table 1 summarizes the description of each server and Table 2 summarizes the network equipment. Table 3 describes in detail the hardware of the IBM eServer Blade .

Table 1 - Servers Description

Role	Architecture	Storage	RAM	Network interfaces	Cores	OS (host)
Master	64 bit	0.5 TB	8 GB	1 Ethernet x 1Gb	8	CentOS 6.4
Controller	64 bit	1 TB	20 GB	3 Ethernet x 1Gb	8	Ubuntu 12.04
Compute 1	64 bit	1 TB	64 GB	3 Ethernet x 1Gb	12	Ubuntu 12.04
Compute 2	64 bit	1 TB	16 GB	3 Ethernet x 1Gb	8	Ubuntu 12.04
Storage	64 bit	2 TB	32 GB	3 Ethernet x 1Gb	8	Ubuntu 12.04

Table 2 - Network Equipment

Manufacturer	Model
HP	V1910-24G Switch JE006A
TP-LINK	TL-SF1016D

Table 3 - IBM eServer Blade Center hardware description

Part Number	Descrição	Qtde
Chassi	**Blade Center H**	
88524TU	IBM eServer BladeCenter(tm) H Chassis with 2x2980W PSU	1
68Y6601	IBM BladeCenter H 2980W AC Power Modules w/Fan Pack	1
46C7191	BNT Virtual Fabric 10Gb Switch Module for IBM BladeCenter	2
39Y9324	Server Connectivity Module for IBM BladeCenter	2
44X1962	Brocade 8 Gb SFP+ SW Optical Transceiver	6
44X1921	Brocade 10-port 8 Gb SAN Switch Module for IBM BladeCenter	2
46M0901	IBM UltraSlim Enhanced SATA DVD-ROM	1
25R5785	2.8m, 200-240V, Triple 16A IEC 320-C20	2
Lâminas HS22V	**HS22V - 144GB RAM**	
7871H2U	HS22V, Xeon 6C X5650 95W 2.66GHz/1333MHz/12MB, 3x2GB, O/Bay 1.8in SAS	2
69Y0924	Intel Xeon 6C Processor Model X5650 95W 2.66GHz/1333MHz/12MB	2
46C0568	8GB (1x8GB, 2Rx4, 1.35V) PC3L-10600 CL9 ECC DDR3 1333MHz VLP RDIMM	36
90Y3550	Emulex 10GbE Virtual Fabric Adapter II - IBM BladeCenter	2
46M6140	Emulex 8Gb Fibre Channel Expansion Card (CIOv) for IBM BladeCenter	2
Lâminas HS22	**HS22 - 96GB RAM (virtualização)**	
7870H2U	HS22, Xeon 6C X5650 95W 2.66GHz/1333MHz/12MB, 3x2GB, O/Bay 2.5in SAS	4
59Y5709	Intel Xeon 6C Processor Model X5650 95W 2.66GHz/1333MHz/12MB	4
46C0568	8GB (1x8GB, 2Rx4, 1.35V) PC3L-10600 CL9 ECC DDR3 1333MHz VLP RDIMM	48
90Y3550	Emulex 10GbE Virtual Fabric Adapter II - IBM BladeCenter	4
46M6140	Emulex 8Gb Fibre Channel Expansion Card (CIOv) for IBM BladeCenter	4
Servidor em rack	**IBM Director Server**	
7944D4U	x3550 M3, Xeon 4C E5620 80W 2.40GHz/1066MHz/12MB, 1x4GB, O/Bay HS 2.5in SA	1
59Y4006	Intel Xeon 4C Processor Model E5620 80W 2.40GHz/1066MHz/12MB	1
49Y1406	4GB (1x4GB, 1Rx4, 1.35V) PC3L-10600 CL9 ECC DDR3 1333MHz LP RDIMM	5
42D0637	IBM 300GB 2.5in SFF Slim-HS 10K 6Gbps SAS HDD	4
59Y3952	IBM System x3550 M3 R2 ODD Kit	1
46M0829	ServeRAID M5015 SAS/SATA Controller	1
68Y7396	ServeRAID M5000 Series Battery Remote Mount Cable	1
46M1076	Dual Port 1Gb Ethernet Daughter Card	1
81Y6558	IBM 460W Redundant Power Supply Unit	1
46M0901	IBM UltraSlim Enhanced SATA DVD-ROM	1
46C7526	IBM Virtual Media Key	1
Storage	**DS3500 Storage**	
1746A4D	IBM System Storage DS3524 Express Dual Controller Storage System	1
49Y2048	600GB 2.5in 10K 6Gb SAS HDD	19
68Y8432	8Gb FC 4 Port Daughter Card	2
39M5697	5m Fiber Optic Cable LC-LC	4
Switches	**BNT Top Of Rack Switch 1/10GB**	
730952F	IBM BNT RackSwitch G8052F	2
90Y9433	5m IBM Passive DAC SFP+ Cable	8
39Y7937	1.5m, 10A/100-250V, C13 to IEC 320-C14 Rack Power Cable	4
Parte Elétrica	**PDUs**	
39Y8935	DPI 63amp/250V Front-end PDU with IEC 309 IEC 309 P+N+G	2
39Y8941	IBM DPI C13 Enterprise PDU w/o Line Cord	2
40K9613	IBM DPI 63a Cord (IEC 309 P+N+G)	2

2.1.3 On-going work

2.1.3.1 *Implementation of the Smart Campus concept*

In several Sao Paulo University campuses (São Paulo City, São Carlos City, Ribeirão Preto City, etc.) we build cooperative action towards the Smart Campus concept. The idea is to aggregate open data already available in the several Institutes, Departments, Museums and Centers of the University of Sao Paulo in order to foster scientific research, public sector policies and innovations.

USP already is implementing smart campus/city concepts in its several campuses. For instance, all the public illumination of the campus is based on LED technology, digital control, and IP addressing. Rain Forest Trees inside the campus ("Mata Atlantica") are tagged by wireless sensor networks and RFIDs. Several Cameras installed inside the campus provide public security information. Cameras installed in the three main gates of the campus monitors the flow of vehicles. In the Sao Carlos Campus, rivers are being monitored in real time by wireless sensor networks. The integration of already existing smart campus and city initiatives in USP campuses will provide opportunity to create innovations to solve common USP and Sao Paulo State Cities problems.

2.1.3.2 *Integration With CloudUSP*

In order to provide a larger infrastructure, is planned the implantation of the USP FIWARE Lab node on top of the USP Cloud infrastructure, shown in detail in Figure 2.

The goal is to provide privileged access to the FIWARE capabilities to the academic community of the University of São Paulo, including professors, students and researchers.

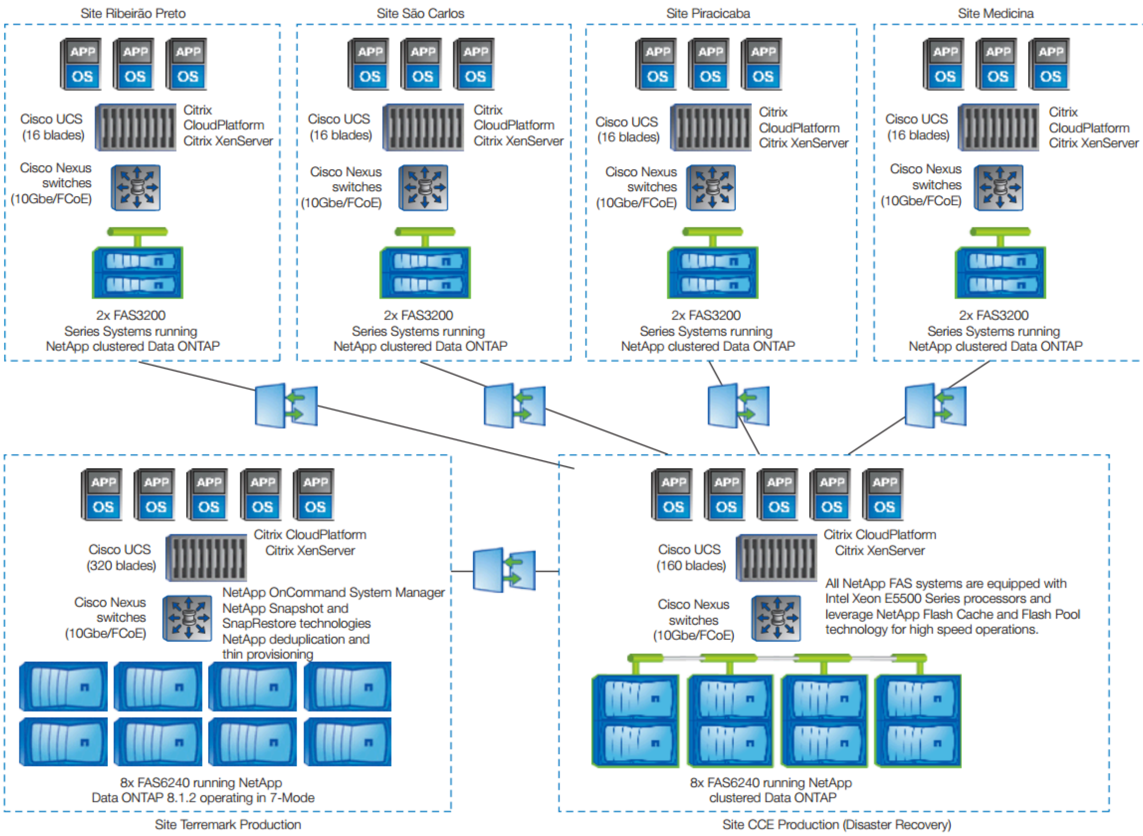


Figure 2- USP Cloud architectural overview

2.1.4 Local Innovation Ecosystem

The USP FIWARE Lab provides the common infrastructure to entrepreneurs, startups and SMEs, mainly in São Paulo State, to develop innovations.

Startups incubated at CIETEC (Centro de Inovação, Empreendedorismo e Tecnologia - www.cietec.org.br), inside the USP São Paulo Campus, benefit directly of the Infrastructure and the technical support of the USP FIWARE Lab.

FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo – www.fapesp.br) has especial funding lines for SMEs in the PIPE ("Pesquisa Inovativa em Pequenas Empresas") and PITE ("Programa de Apoio à Pesquisa em Parceria para Inovação Tecnológica") Funding Programs.

Public Sector (São Paulo City, São Paulo State, and other cities) as well as large São Paulo State demand solutions, which can use Future Internet Technologies (Internet of Things, Smart City, Smart Grid, Smart Utilities such as water and gas, public transportation, public security, emergency and disaster prevention, large events, etc.

2.2 Regional Center at Uberlândia (UFU)

In this section, we describe the infrastructure deployed for the FIWARE Regional Center at UFU. The node was deployed at Campus Santa Mônica, located at Uberlândia, Minas Gerais state in Brazil. Figure 3 presents the campus deployment and the building where related infrastructure was installed.

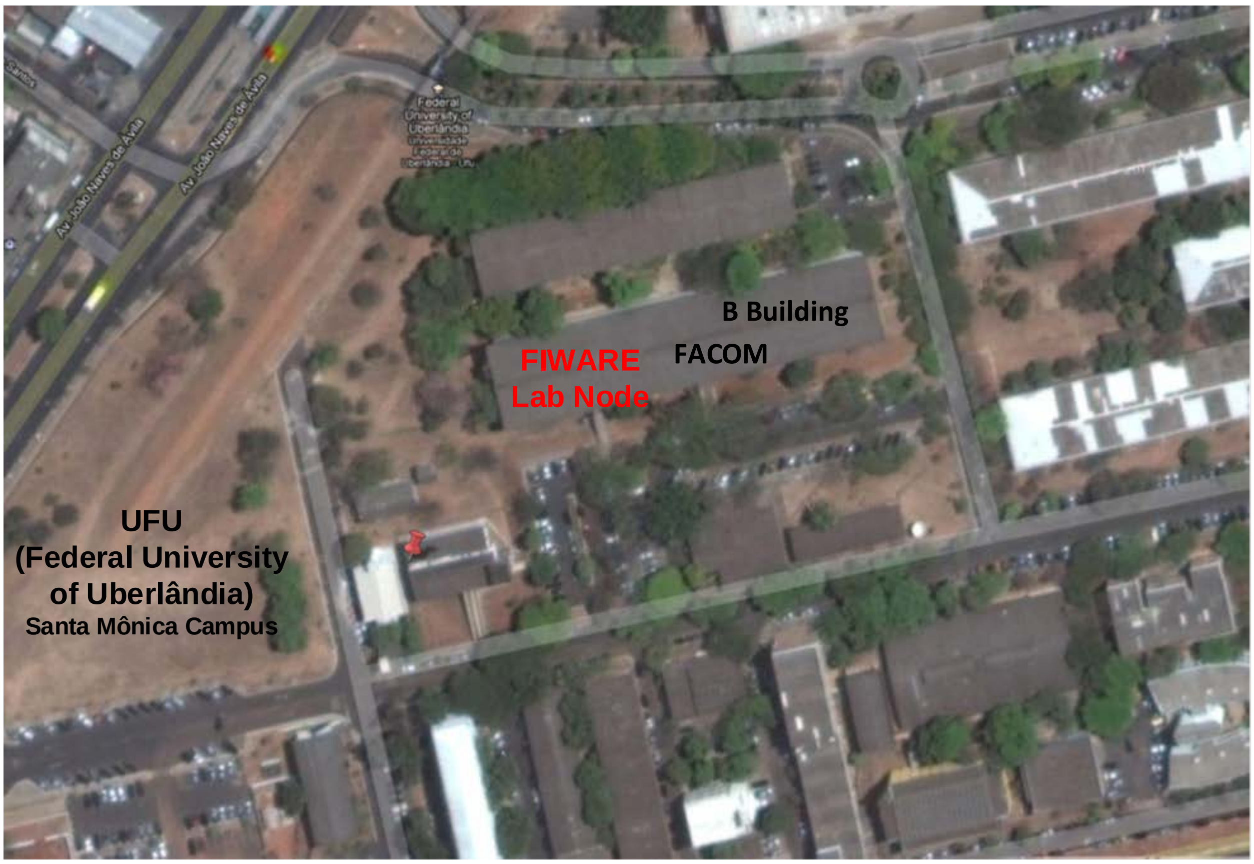


Figure 3 - Campus Deployment at Uberlândia

2.2.1 Infrastructure

Two racks were used to accommodate the servers and the switch (TOR Switch) that compose the infrastructure. Figure 4 presents an overall diagram of the FIWARE racks and the equipment that was installed on it. Each server is connected to four different networks, using two different network cards. The logical networks are:

- Management – Using VLAN 101 (tagged) at eth0. IP range: 200.19.151.130 to 200.19.151.145;
- Storage – Using VLAN 102 (tagged) at eth0. IP range: 192.168.1.0/24;
- Public/Floating – Using VLAN 100 (tagged) at eth1. IP range: 200.19.151.146 to 200.19.151.180;
- Private – Using VLAN 103 (tagged) at eth1. IP range: 10.0.0.0/24.

Table 4 presents the hardware inventory of the servers deployed at the FIWARE node at UFU. It shows the role of each server and describes the main characteristics of each one considering processor, memory and storage. Table 5 details the TOR Switch connections and configuration at each port as presented in the diagram displayed at Figure 4. Finally, Figure 5 show some pictures of the local site of UFU, presenting details of the deployed hardware.

.

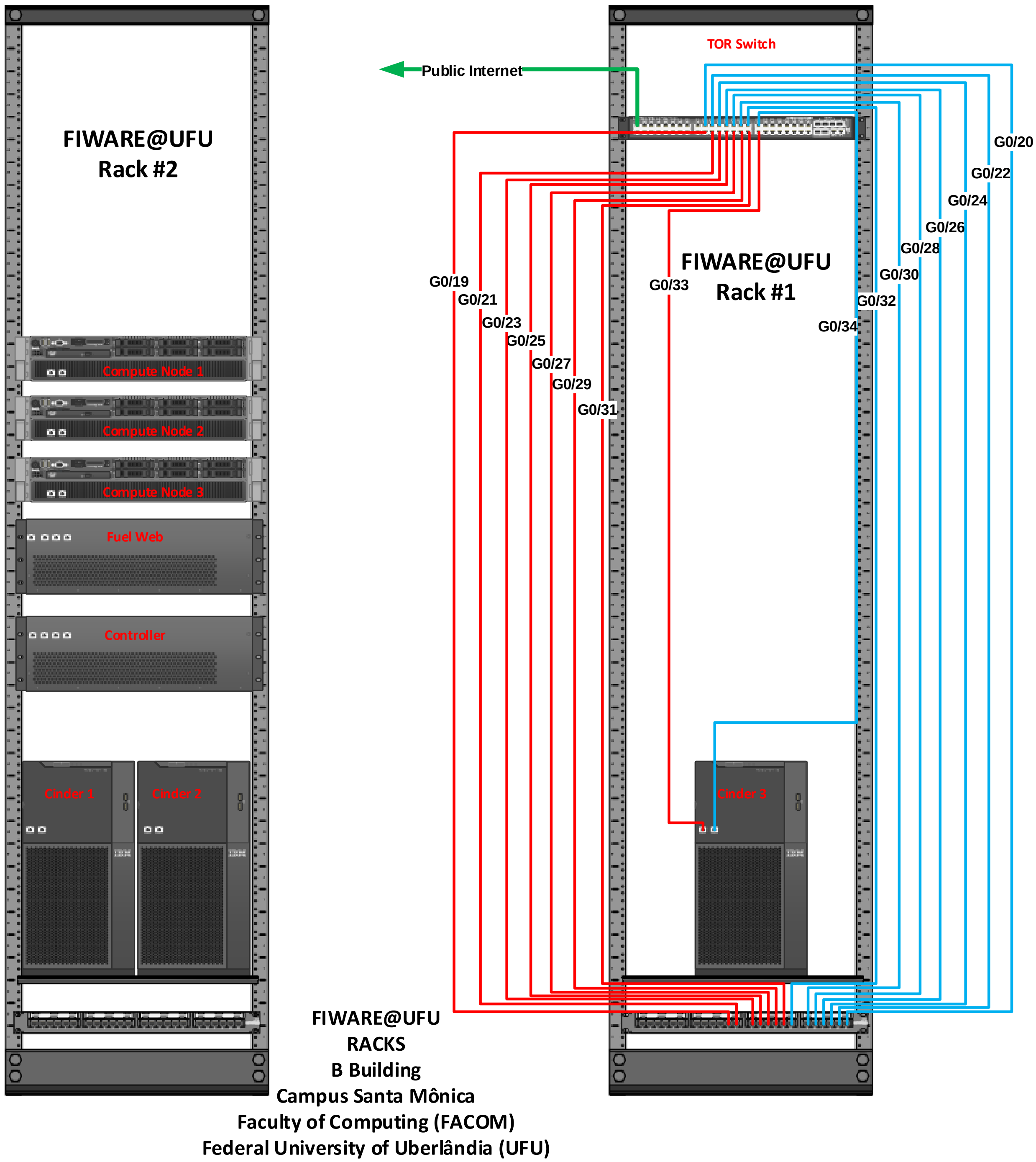


Figure 4 - FIWARE Racks at UFU

Table 4 - Hardware Inventory Deployed at UFU

ROLE	MANUFACTURER	MODEL	PROCESSOR TYPE	NUMBER OF PROCESSORS	NUMBER OF CORES	TOTAL CORES	MEMORY (GB)	TOTAL STORAGE (GB)
ITBox	Intel	Not Applicable	E5-2630	2	6	12	64	2000
Controller	Intel	Not Applicable	E5-2650v2	2	8	16	96	2000
Compute Node 1	HP	Proliant DL380 G5	E5440	2	4	8	32	500
Compute Node 2	HP	Proliant DL380 G5	E5440	2	4	8	32	500
Compute Node 3	HP	Proliant DL380 G5	E5405	2	4	8	24	1000
Cinder 1	IBM	x3500	E5405	2	4	8	24	500
Cinder 2	IBM	x3500	E5440	2	4	8	32	500
Cinder 3	IBM	x3500	E5405	2	4	8	24	500

Table 5 - TOR Switch Configuration

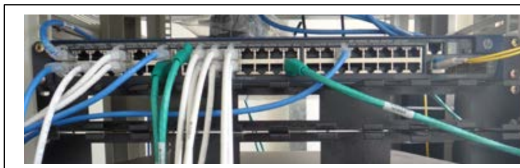
SWITCH PORT	SERVER ROLE	SERVER NIC	TYPE	VLAN ID
G0/19	ITBox	eth0	untagged	103
G0/20	ITBox	eth1	untagged	100
G0/21	Controller	eth0	tagged	101, 102
G0/22	Controller	eth1	tagged	100, 103
G0/23	Compute Node 1	eth0	tagged	101, 102
G0/24	Compute Node 1	eth1	tagged	100, 103
G0/25	Compute Node 2	eth0	tagged	101, 102
G0/26	Compute Node 2	eth1	tagged	100, 103
G0/27	Compute Node 3	eth0	tagged	101, 102
G0/28	Compute Node 3	eth1	tagged	100, 103
G0/29	Cinder 1	eth0	tagged	101, 102
G0/30	Cinder 1	eth1	tagged	100, 103
G0/31	Cinder 2	eth0	tagged	101, 102
G0/32	Cinder 2	eth1	tagged	100, 103
G0/33	Cinder 3	eth0	tagged	101, 102
G0/34	Cinder 3	eth1	tagged	100, 103
G0/2	Router (default gateway)	NA	untagged	100

(a) Full View of Servers at Rack #2

(b) Compute Servers

(c) Fuel and Controller Servers

(d) Cinder Servers at Rack #2



(d) TOR Switch During Deployment

Figure 5 – Deployed Hardware

2.2.2 Local Ecosystem

Each Regional Center is linked to the local innovation ecosystem. At Uberlândia, i9NAGI (http://www.i9nagi.ufu.br/) and CIAEM (http://www.ciaem.org.br/) are two of the partners that are committed to the regional center.

i9NAGI is an innovation support management center that aim to develop innovation and entrepreneurship with the local software industry of Uberlandia. Funded by FINEP, the i9NAGI center has thirty-five local ICT companies from different sizes.

CIAEM is an incubation center of entrepreneurial activities, managed and located inside UFU. The mission of CIAEM is to encourage the creation and development of new technology-based business by supporting young companies with infrastructure and as a business advisor helping them to create their business plan. Although at this moment CIAEM support was not required, they are ready to host a new company that can explore marketing opportunities with an ICT solution based on the FIWARE platform. In this case, the Regional Center at UFU offers the technical infrastructure and CIAEM, provides the support in the business field.

Two other entities from the local innovation ecosystem are stakeholders of the Regional Center at UFU. SEBRAE (www.sebraemg.com.br/atendimento) and i9Uberlandia (http://www.i9uberlandia.org.br/).

SEBRAE supports micro and small enterprises. SEBRAE is a non-profit private entity with the mission of promoting the sustainable and competitive development of small businesses. At Uberlandia SEBRAE supports several ICT companies through the Polo Mobility, a project created by SEBRAE-MG in order to improve the performance of software companies Uberlândia through technology consulting, managerial and technical skills, attracting and generating new business, and the creation of strategic partnerships to strengthen local governance, making the city renowned for creating mobile solutions. Today, about sixty technology companies of Uberlandia participate in the Polo Mobility. Those companies have created software and applications to meet slices of the business market with better results.

i9Uberlandia is an association of local ICT companies from Uberlândia. Its main goal is the development and promotion of the information technology sector in the region, supporting businesses in the form of technical and management training, business generation and supporting the development of projects for funding, aimed at increasing the participation of these companies in the regional market.

3 Proof of Concepts

The proof of concepts created inside this task has two main objectives: the first one was to highlight FIWARE platform focusing local challenges, and the second one is to be used as an instrument for the evangelism and training of local stakeholders. This section describes two different applications created in each regional center: one at São Paulo and another at Uberlândia.

3.1 São Paulo

Cloud hosting landscape has changed dramatically in the last few years. Application providers can choose among large variety of options to host their applications and data – both in terms of hosting abstraction (e.g., virtual machine versus Java container) as well as various non-functional specifications and business models (e.g., private vs public).

In order to measure in more accurate way the living quality in the urban space, we proposed a technological architecture to support data collection for mobile crowd sensing from mobile device sensors, prototyping board sensors, wearable devices and open data sources. The FIWARE infrastructure was used to deploy the Cloud Computing capabilities. The main contribution of this work is to associate data from different sources to supply citizens with a more precise real-time feedback on the environment conditions of the city.

SmartCampus is set of physical devices –sensors and wearable devices– and Cloud-based software prototype that measure the contamination levels, air quality and acoustic pollution in the University of São Paulo Campus (USP) with smartphones and sensors (Android smartphones and Arduino sensors). The final goal is to perform these measures over the public transport system of São Paulo city (buses and metro). SmartCampus makes use of several FIWARE building blocks (in the form of Generic Enablers - GEs), all of them deployed in the FIWARE Lab testbed, as it depicted in Figure 6:

- Orion Context Broker: a data concentrator for all the measures; manages subscriptions and data access. The Orion Context Broker has to be integrated to the Cosmos enabler in order to achieve persistent/historical storage.
- Wirecloud: Front End based on HTML5+JS widget composing, which offers some libraries for integrating the Context Broker.
- Cosmos: Big Data platform to store and analyze measure data.

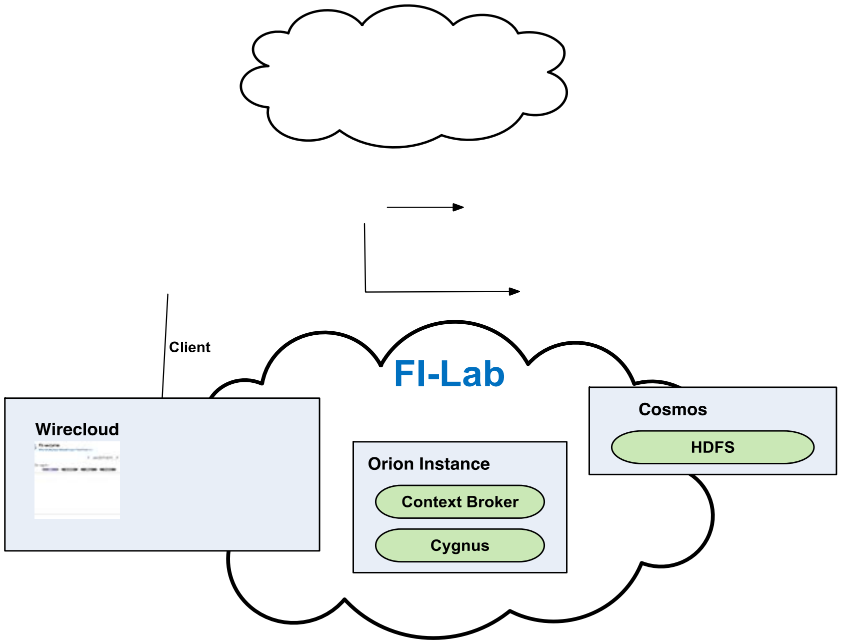


Figure 6 - SmartCampus Framework Architecture

All the sensor-reading activities are done in Android smartphones that receive all information sent by Arduino sensors. Then the Android application sends all the collected data to the Context Broker using the NGSI Protocol.

The map viewer uses data from the Context Broker in order to show a map of the campus with the real-time information about temperature (signaling what parts are too cold or too warm), noise, pressure, etc., as well some other information from the internal sensors of the mobile device, as illustrated in Figure 7.

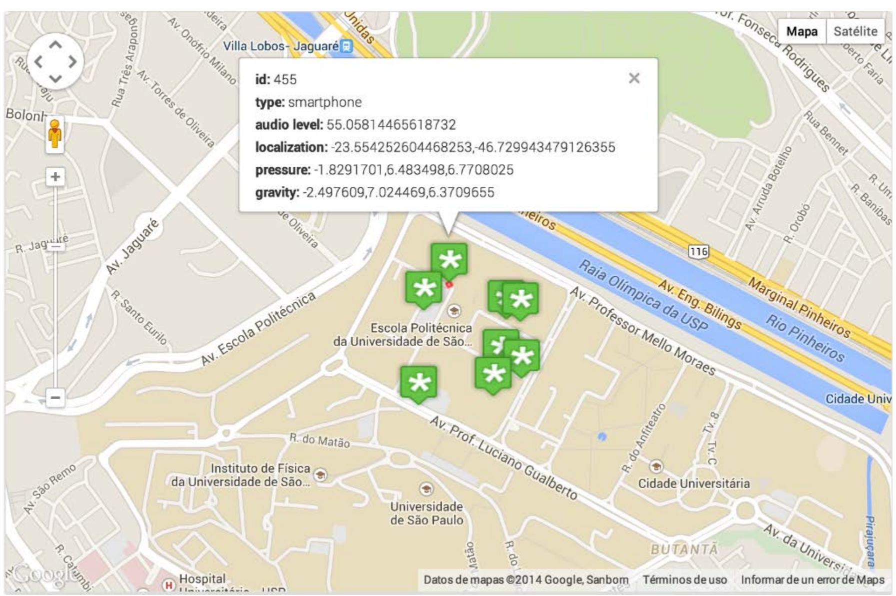


Figure 7 - Web Interface to SmartCampus Application

3.1.1 Android Application

The mobile application uses resources available in the Android API to perform the monitoring and collecting of data from internal android device sensors. Additionally, the Android Application can serve as a bridge to send data gathered by Arduino and Galileo boards.

The application gathers data from all sensors of the mobile device, such as user location, noise degree, brightness level, accelerometer and gyro data and so on. To preserve user privacy, no personal data is gathered. The application runs as a service, in the background, but user has the option to stop the service at any time.

Android is a software environment built for mobile devices; hence, it is not a hardware platform. Android includes a Linux Kernel-based OS, a rich user interface, end-user applications, code libraries, application frameworks, multimedia support, etc. User applications are built for Android in Java. Even the built-in applications are written in Java. There is no difference between the built-in applications and applications created with the SDK.

Figure 8 presents an overview of the application architecture. The application consists of the following packages: **`main`**, **`util`** and **`rw`**. The last one has two sub-packages named readers and writer. The package main has essentials classes to application. **`SmartCampusAct`** is the class that represents the user interface and **`SmartService`** is the service responsible to collect data from sensors. The package **`util`** has a set of constants. Each of them is encapsulated in interfaces, classes and enumerators. The package **`rw`** has classes responsible for collect data from smartphone or Arduino boards. This work is done through classes stored in the reader package. Additionally, **`rw`** has the class responsible for sending data to the FIWARE platform and the **`ControllerRW`** class that is responsible for the creation and control of classes in the package.

Figure 9 shows our application's Sequence Diagram. Two actors are displayed in the diagram: the smartphone user and the FIWARE server. The smartphone user starts the application by clicking on launch icon. This action creates an object of type SmartCampusAct. The **`SmartCampusAct`** graphical class is responsible for creating the **`SmartService`** object by calling the **`startService()`** method. **`SmartService`** instantiates a **`ControllerRW`** object that is responsible for controlling objects that gather data and send them to the FIWARE server. The work of gathering and sending data is done through the readers classes (**`ArduinoReader`**, **`GeoLocReader`**, **`AudioLevelReader`** and **`SensorsReader`**) and the writer class WriterJSON, respectively.

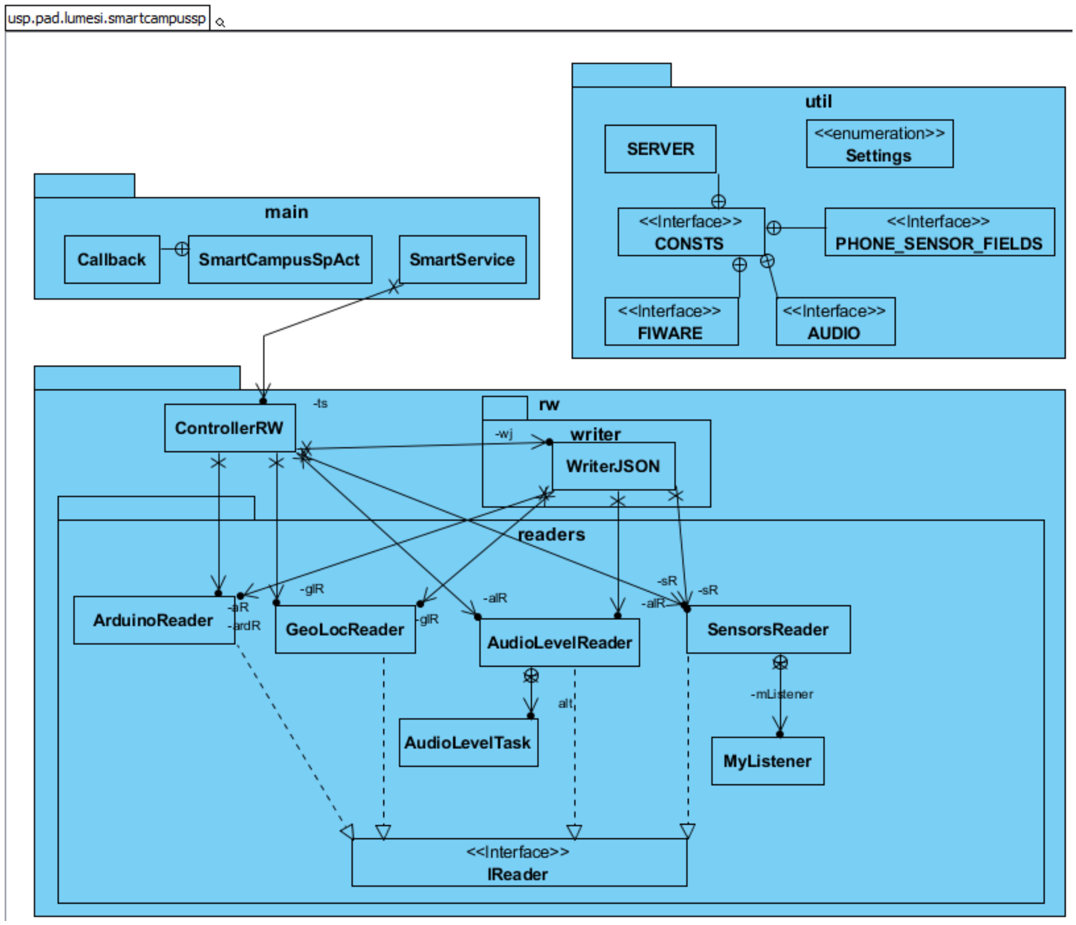


Figure 8 - Android Application Package Diagram

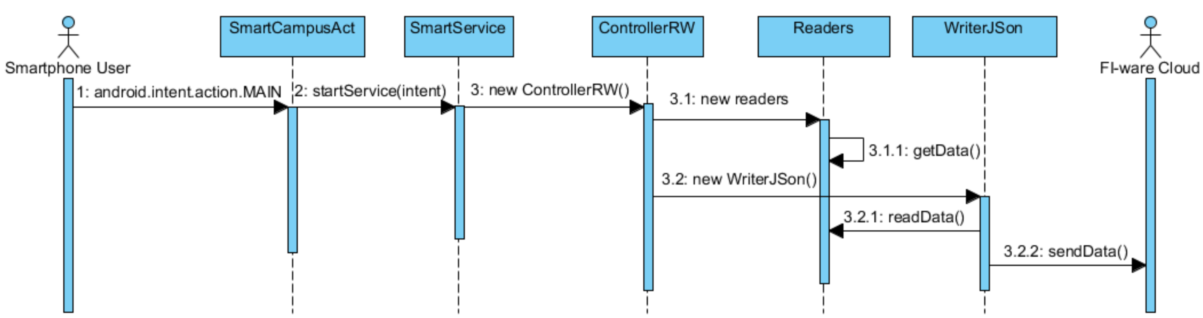


Figure 9 - Android Application Sequence Diagram

In the next paragraphs, we detail each one of those components. First, for a better understanding, it is necessary to introduce some components of the Android Platform. The Android platform has the following main components to build applications: **`Activity`**, **`Service`**, **`BroadcastReceiver`** and **`ContentProvider`**.

Activity represents the graphical component in an Android application. Usually, android applications have more than one **Activity**. Each **Activity** displays a user interface and responds to system and user-initiated events. An **Activity** consists in one or more **Views** to present the user interface elements. Figure 10 shows an example of an **Activity**. This activity corresponds to the **SmartCampusSpAct**, the **Activity** present in our Android application. This activity allows the user to configure the data collection period and what are the sources of data: Arduino sensors, smartphone sensors or both. Additionally, the activity is responsible for start or stop the data gathering service through the Start Service and Stop Service buttons, respectively.

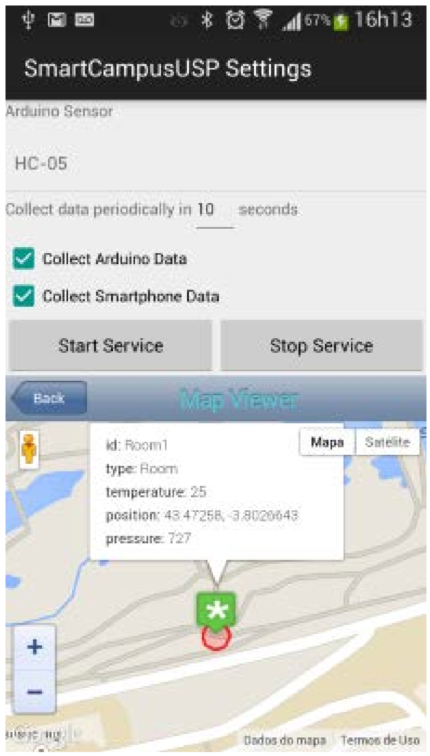


Figure 10 - Graphical view of SmartCampusSpAct

The **Activity** class is part of the **android.app** Java package, found in the Android runtime. The Android runtime is deployed in the **android.jar** file. One of the primary tasks an **Activity** performs is displaying UI elements, which are implemented as Views and are typically defined in XML layout files.

Listing 1 shows the **onCreate ()** method of the **SmartCampusSpAct**. This method is called at the time when the Activity is created. Line 40 calls the **setContentView()** method. It indicates that the layout and graphics that make up this **Activity** are described in the file **activity_main.xml**.

```
1  package usp.pad.lumesi.smartcampussp.main;
2
3  import java.util.ArrayList;
33
34 public class SmartCampusSpAct extends Activity {
35
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         //default init calls
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_main);
41         intent = new Intent("br.usp.lumesi.INIT_SMART");
42         getInterfaceElementsReferences();
43         recreateUI(savedInstanceState);
44         createListeners();
45         initBluetooth();
46         getPreferences();
47     }
```

Listing 1 - The onCreate() method of the SmartCampusSpAct

In line 41, it is instantiated a new **`Intent`**. An **`Intent`** is a declaration of need. It is made up of a number of pieces of information that describe the desired action or service. The action attribute "**`br.usp.lumesi.INIT_SMART`**" is used to start the service **`SmartService`**. The action attribute of an **`Intent`** is typically a verb: for example **`VIEW`**, **`PICK`**, or **`EDIT`**. A number of built-in Intent actions are defined as members of the Intent class, but application developers can create new actions as well. When an Intent is dispatched, the system evaluates the available Activities, Services, and registered BroadcastReceivers and dispatches the Intent to the most appropriate recipient.

Line 42 calls the **`getInterfaceElementsReferences()`** method that combines class attributes references to graphic objects present in the **`Activity`**. The methods **`recreateUI()`** and **`getPreferences()`** (lines 43 and 46) retrieves configuration values modified in past executions.

The behavior of graphical components present in the Activity is given by **`createListeners()`** method (line 44). On Android, we can associate classes to graphical components. Thus, when an action is performed on this component, the listener class is responsible for their treatment. Line 45 calls the **`initBluetooth()`** method responsible for turning on the Bluetooth module to perform communication with Arduino devices.

Listing 2 shows a snippet of the activity_main.xml file. In Listing 2, we see that for the construction of **`SmartCampusSpAct`** was used **`LinearLayout`** layout manager (lines 1-10). **`LinearLayout`** distributes graphical components linearly. This distribution can be horizontal or vertical. Lines 11-16 show the **`TextView`** declaration. A **`TextView`** is a graphical component that represents a label. The value of a **`TextView`** is passed through the android:text property. In this example, the **`TextView`** has the value Arduino Sensor.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="usp.pad.lumesi.smartcampussp.writer.SmartCampusSpAct" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:orientation="horizontal" >
        <TextView
            android:id="@+id/txtArduino"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Arduino Sensor"
            android:textSize="16sp" />
```

Listing 2 - A snippet of the activity_main.xml file

Listing 3 shows a snippet of **createListeners** method. Lines 50 to 69 associate an **OnClickListener** object to a **btnSettings** button. By clicking the button **btnSettings**, the values of text box **edtTime** and the check box **chkES** and chkIS are captured and saved to an intent object (lines 56-58). Additionally the value selected in **ListView lvBlue** is used. **lvBlue** displays a list of nearby devices Arduino compatible with our Android app. It is then called the **startService()** method that starts the service SmartService passing as a parameter the intent object.

```
49      private void createListeners(){
50          btnSettings.setOnClickListener(new OnClickListener() {
51
52              @Override
53              public void onClick(View v) {
54                  try{
55                      String time = edtTime.getText().toString();
56                      intent.putExtra(Settings.TIME.name(),
57                              time.length()==0?CONSTS.DEFAULT_TIME_VALUE:Integer.parseInt(time)*1000);
58                      intent.putExtra(Settings.ENABLE_EXTERNAL_SENSORS.name(), chkES.isChecked());
59                      intent.putExtra(Settings.ENABLE_INTERNAL_SENSORS.name(), chkIS.isChecked());
60                      int pos = lvBlue.getCheckedItemPosition();
61                      CONSTS.SERVER.BLUETOOTH_SERVER = lvBlue.getItemAtPosition(pos).toString();
62                      startService(intent);
63
64                      finish();
65                  } catch (NullPointerException npe){
66                      print("Click on device name and then click on start service");
67                  }
68              }
69          });
```

Listing 3 - Snippet of the createListeners() method from SmartCampusSpAct class

If an application is to have a long lifecycle, it is often best to put it into a Service. For example, a background data-synchronization utility should be implemented as a Service. A best practice is to launch Services on a periodic or as-needed basis, triggered by a system alarm, and then have the Service terminate when its task is complete. Similarly to the Activity, a Service is a class in the Android runtime that can be extend.

Building a service requires that the package **android.app.Service** is imported. This package contains the Service class. The **SmartService** class extends the Service class. The **onCreate ()** method of the Service class permits the application to perform initialization-type tasks. Services are started with the **startService(Intent)** method of the abstract Context class. Listing 4 shows the o**nCreate()** method of the **SmartService**. This method creates a new item on the Android notification bar (lines-41-48). Additionally, puts the service to run in the foreground (line 54), in order to avoid the service to be suspended for Android garbage collector. Figure 11 presents the changes made to the Android notification bar.

```
17 public class SmartService extends Service {
18     private int timeInterval;
19     private ControllerRW ts;
20
22     public IBinder onBind(Intent intent) {
26
27
28     @Override
29     public void onCreate() {
30
31         // Create an Intent that will open the main Activity
32         // if the notification is clicked.
33         int NOTIFICATION_ID = 1;
34         Intent intent = new Intent(this, SmartCampusSpAct.class);
35         PendingIntent pi = PendingIntent.getActivity(this, 1, intent, 0);
36         Intent stopIntent = new Intent(this, SmartService.class);
37         stopIntent.setAction("ups.pad.lumesi.STOP");
38         PendingIntent pstopIntent = PendingIntent.getService(this, 0,
39                 stopIntent, 0);
40         // Set the Notification UI parameters
41         Notification notification = new Notification.Builder(this)
42         .setContentTitle("SmartCampusSp")
43         .setTicker("SmartCampusSp")
44         .setContentText("SmartService")
45         .setContentIntent(pi)
46         .setOngoing(false)
47         .addAction(android.R.drawable.ic_media_pause, "Stop Service", pstopIntent)
48         .setSmallIcon(R.drawable.ic_launcher).build();
49
50         notification.flags = notification.flags |
51                         Notification.FLAG_ONGOING_EVENT;
52
53         // Move the Service to the Foreground
54         startForeground(NOTIFICATION_ID, notification);
55     }
56
```

Listing 4 - The onCreate() method of SmartService

After the system retrieves the service (creating it and calling its **onCreate()** method), system calls its own **onStartCommand(Intent, int, int)** method with the arguments supplied by the client. At this point, the service will continue running until **Context.stopService()** or **stopSelf()** is called. Multiple calls to **Context.startService()** do not nest (though they do result in multiple corresponding calls to **onStartCommand()**).

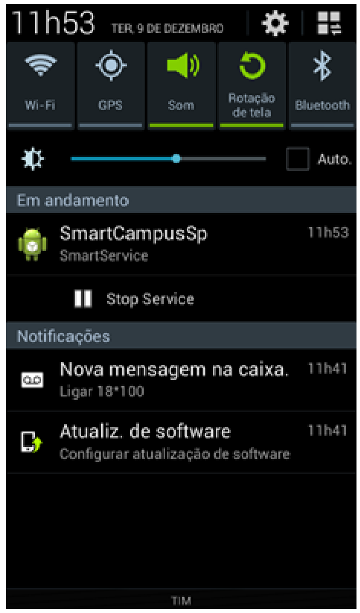


Figure 11 - Notification Bar with shortcut to SmartCampusSp Application

Listing 5shows the implementation of the method **`onStartCommand()`** of **`SmartService`**. Line 63 calls the **`ConfigService()`** responsible for configuring the service class. Depending on the last intent, **`ConfigService`** can generate a new **`ControllerRW`** object (line 72), it can stop all threads initialized by **`ControllerRW`** and stop the **`SmartService`** service (lines 74-75) or create a new **`ControllerRW`** object, but modifying the period for collecting data (lines 77-79).

```
58    @Override
59    /**
60     * Start SmartService. Use intent to configure service.
61     */
62    public int onStartCommand(Intent intent, int flags, int startId) {
63        configService(intent);
64        Toast.makeText(getBaseContext(), "Service running",
65                Toast.LENGTH_SHORT).show();
66        return START_STICKY;
67    }
68
69
70    private void configService(Intent intent){
71        if (intent == null){
72            ts = new ControllerRW(getDefaultIntent(), getBaseContext());
73        } else if (intent.getAction().equals("ups.pad.lumesi.STOP")){
74            ts.stop();
75            stopSelf();
76        } else{
77            Log.d(CONSTS.TAG, CONSTS.SERVER.BLUETOOTH_SERVER);
78            timeInterval = intent.getExtras().getInt(Settings.TIME.name(),
79                    CONSTS.DEFAULT_TIME_VALUE);
80            ts = new ControllerRW(intent, getBaseContext());
81        }
82
83    }
```

Listing 5- onStartCommand() method of SmartService

Listing 6 shows the source code of **ControllerRW** class. ControllerRW is a control class responsible for creating all sensor readers. In our application, there are the following readers: **SensorsReader**, **ArduinoReader**, **AudioLevelreader** and **GeolocReader**. The **ArduinoReader** is responsible for forwarding the data gathered from Arduino or Galileo board located next to the smartphone. **AudioReader** and **GeolocReader** are responsible for reading the noise degree and user location, respectively. Finally, the **SensorReader** is responsible for reading data from internal sensors of the smartphone. **ControllerRW** also creates the **WriterJSON** Object. **WriterJSON** encapsulates data in a JSON package and send them to the FIWARE server. Additionally, the stop method is responsible for interrupting the execution of threads of some reader classes, and the thread of the writer class (lines 35-42).

```
13 public class ControllerRW{
14     private Intent settings;
15     private SensorsReader sR;
16     private ArduinoReader aR;
17     private AudioLevelReader alR;
18     private GeoLocReader glR;
19     private WriterJSON wj;
20
21     public ControllerRW(Intent settings, Context ctx) {
22         this.settings = settings;
23         int time = settings.getExtras().getInt(Settings.TIME.name(), CONSTS.DEFAULT_TIME_VALUE);
24         if (settings.getExtras().getBoolean(Settings.ENABLE_EXTERNAL_SENSORS.name(), false))
25             this.aR = new ArduinoReader(time);
26         if (settings.getExtras().getBoolean(Settings.ENABLE_INTERNAL_SENSORS.name(), false)){
27             this.sR = new SensorsReader(ctx, time);
28             this.glR = new GeoLocReader(ctx, time);
29             this.alR = new AudioLevelReader(time, CONSTS.AUDIO.FAST_MODE);
30         }
31         this.wj = new WriterJSON(time, aR, alR, glR, sR, ctx);
32     }
33
34     public void stop(){
35         if (settings.getExtras().getBoolean(Settings.ENABLE_EXTERNAL_SENSORS.name(), false))
36             aR.stop();
37         if (settings.getExtras().getBoolean(Settings.ENABLE_INTERNAL_SENSORS.name(), false)){
38             alR.stop();
39             sR.stop();
40             glR.stop();
41         }
42         wj.stop();
43     }
44 }
```

Listing 6 - ControllerRW class

Listing 7 shows the code snippet of the **geoLocReader**. The **GeoLocReader** class is responsible for performing the gathering of the user's location through the GPS module present in this Smartphone). **GeolocReader** creates an instance of **LocationListener** object (lines 87-116) that is sensitive to the user's location changes. At the time when the user changes his or her location, **onLocationChanged** method is called. In the body of the method, the new values for the location are saved in variable values (lines 112-115)

```
package usp.pad.lumesi.smartcampussp.readers;

import java.util.Locale;

public class GeoLocReader implements IReader {
    private void createListeners(){
        ll = new LocationListener() {

            @Override
            public void onLocationChanged(Location location) {
                values[0] = location.getLongitude();
                values[1] = location.getLatitude();
                values[2] = location.getAltitude();
                values[3] = location.getSpeed();
            }
```

Listing 7 - Code snippet of the class GeolocReader

Listing 8 shows a snippet of **WriterJSON** class. **WriterJSON** has a thread that periodically, encapsulates the data collected by sensors in **json** objects (line 38) and then sends this data to the FIWARE server (line 39). The **createJSONMsg** method is responsible for creating the JSONObject objects. Note that each of the readers has a method for creating **JSONObject** (lines 56-59).

The component **BroadCastReceiver** can be employed if the application wants to receive and respond to a global event, such as a ringing phone or an incoming text message. **ContentProvider**, is used when an application manages data and needs to expose that data to other applications. **BroadCastReceiver** and **ContentProvider** are not used in our application. Because of that, we will not detail such components.

All those pieces of information need to be tied together for an Android application to execute properly. The glue mechanism for this task of defining relationships is the **AndroidManifest.xml** file. The **AndroidManifest.xml** file exists in the root of an application directory and contains all the design-time relationships of a specific application and Intents. AndroidManfest.xml files act as deployment descriptors for Android applications.

```java
public class WriterJSON {

    private void createThreadSender(){
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                try {
                    JSONObject jsonMsg = createJSONFIWAREMsg();
                    HttpResponse rsp = sendMessage(jsonMsg);
                    for (Header str : rsp.getAllHeaders())
                        Log.d(CONSTS.TAG, "Status:  " + str);
                } catch (JSONException e) {
                    Log.d(CONSTS.TAG, "Status: Error JSON  " );
                } catch (ClientProtocolException e) {
                    Log.d(CONSTS.TAG, "Status:  Error ClientProtocol" );
                    e.printStackTrace();
                } catch (IOException e) {
                    Log.d(CONSTS.TAG, "Status: Error I/O");
                    e.printStackTrace();
                }
            }
            private JSONObject createJSONMsg() throws JSONException{
                JSONObject jsonMsg = new JSONObject();
                JSONObject jPhoheChild = new JSONObject();
                JSONObject jSensorsChild = new JSONObject();
                glR.inserValues(jPhoheChild);
                sR.inserValues(jPhoheChild);
                alR.inserValues(jPhoheChild);
                ardR.inserValues(jSensorsChild);
                jsonMsg.put(CONSTS.PHONE_SENSOR, jPhoheChild);
                jsonMsg.put(CONSTS.ARDU_SENSOR, jSensorsChild);
                return jsonMsg;
            }
```

Listing 8 - Code snippet of the class WriterJSON

Listing 9 shows an example of the AndroidManifest.xml The manifest element contains the obligatory namespace, as well as the Java package name containing this application (lines 2-3). This application contains a single Activity, with the class name **`SmartCampusSpAct`** (line 18). Note also the @string tag. Any time an @ symbol is used in an AndroidManifest.xml file, it references information stored in one of the resource files (lines 14-16, 19 and 28-29). In this case, the label attribute is obtained from the string resource identified as app_name. This application's alone Activity contains a single **`IntentFilter`** definition. The **`IntentFilter`** herein employed is the most common **`IntentFilter`** seen in Android applications. The action **`android.intent.action.MAIN`** indicates that this is an entry point to the application. The category **`android.intent.category.LAUNCHER`** It is possible to have multiple Activity elements in a manifest file (and thereby an application), with zero or more of them visible in the launcher window. In addition to the elements used in the sample manifest file, other common tags are as follows:

- The <service> tag (lines 25-29) represents a Service. The attributes of the <service> tag include its class and label. A Service might also include the <intent-filter> tag.
- The <receiver> tag represents a BroadcastReceiver, which might have an explicit <intent-filter> tag.
- The <uses-permission> tag tells Android that this application requires certain security privileges (lines 6-12). For example, if an application requires access to the contacts on a device, it requires the following tag in its AndroidManifest.xml file:
    - <uses-permission android:name="android.permission.READ_CONTACTS" />

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="usp.pad.lumesi.smartcampussp"
4      android:versionCode="1" android:versionName="1.0" >
5      <uses-sdk android:minSdkVersion="16" android:targetSdkVersion="21" />
6      <uses-permission android:name="android.permission.BLUETOOTH"/>
7      <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
8      <uses-permission android:name="android.permission.RECORD_AUDIO" />
9      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
10     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
11     <uses-permission android:name="android.permission.INTERNET"/>
12     <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
13     <application android:allowBackup="true"
14         android:icon="@drawable/ic_launcher"
15         android:label="@string/app_name"
16         android:theme="@style/AppTheme" >
17         <uses-library android:name="com.google.android.maps"/>
18         <activity android:name="usp.pad.lumesi.smartcampussp.writer.SmartCampusSpAct"
19             android:label="@string/app_name">
20             <intent-filter>
21                 <action android:name="android.intent.action.MAIN" />
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25         <service android:enabled="true"
26             android:exported="true"
27             android:name="usp.pad.lumesi.smartcampussp.main.SmartService"
28             android:icon="@drawable/ic_launcher"
29             android:label="@string/service_name">
30          <intent-filter>
31             <action android:name="br.usp.lumesi.INIT_SMART" />
32         </intent-filter>
33             </service>
34     </application>
35 </manifest>
```

Listing 9 - Code Snippet of the AndroidManifest.xml

3.2 Uberlândia

The proof of concept of developed at Uberlândia focused on a real local problem. One mobile application, called “Geo-SIT”, was not working properly. This goal of such application was to show in real time position information about the public buses at the city. Focusing this problem, we captured the opportunity and proposed the development of a mobile application based of the FIWARE platform that would solve this problem. The application integrates with open data from traffic office of the municipality of Uberlândia.

Figure 12, briefly presents the main architecture of the application. The **`IntegrationController`** module is responsible to retrieve the the open data provided by the municipality of Uberlândia. Then by calling NSGI10 **`updateContext`** calls it maintains the context stored the **`Orion Context Broker.`**

Cygnus subscribes to Orion and as long the buses information are updated, a notification is sent to Cygnus that is responsible to store the data in a CKAN server. The data stored in CKAN will be available in the FIWARE node at Uberlândia.

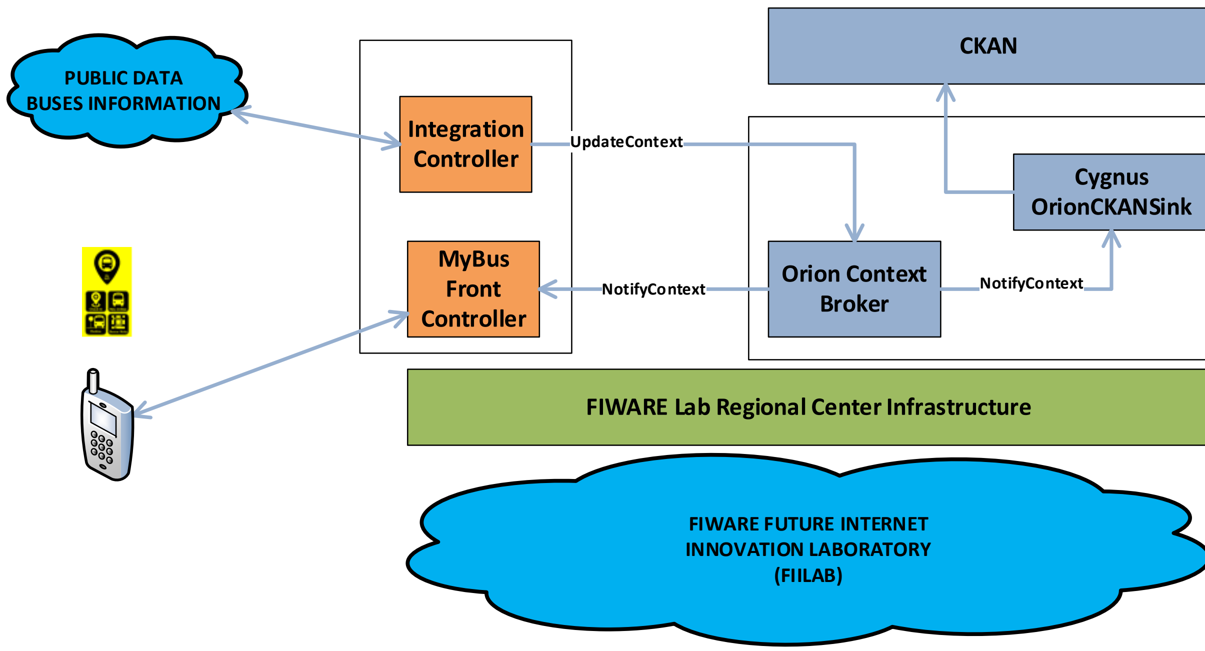


Figure 12- MyBus Application Architecture

The application, called MyBus, is based on the context of Smart City, focused on urban mobility. Figure 13presents some screen of the application. A citizen that intend to take a public bus starts the application and informs the name of the intended bus. After that, a query at the **`MyBusFrontController`**, that is responsible for the communication with the mobile device, will bring the route and the position of all the buses that are currently in operation. Then the application will update the information and the user will track the position.

The development was conduct by the UFU's team and the application will be offered to all citizens of Uberlândia (about 250K daily users of the public transportation). Since the buses equipment's are about to be updated, currently the application is not using real data. Thus, instead of offering partial information regarding the buses position, the traffic office of Uberlândia prefers conclude such update on the buses before publicly offering the buses position data. . However, interfaces were planned in order to ease the integration of the MyBus with the data sources of the buses position information.

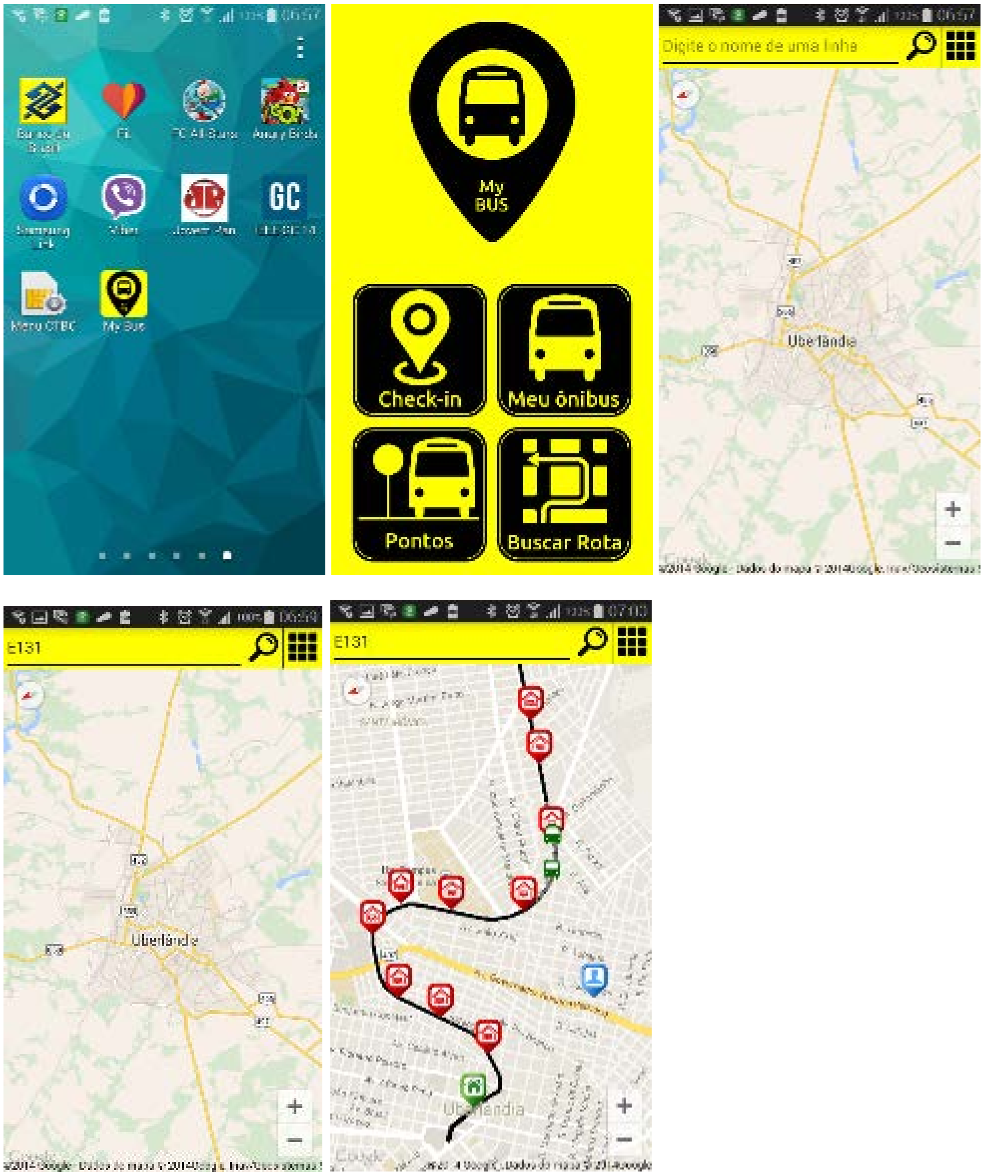


Figure 13 - MyBus Screenshots

4 Evangelism, Training and Dissemination

Several actions regarding evangelism, training and dissemination were conducted by USP and UFU. This section describes these actions.

4.1 Workshop "Contribuições para Cidades inteligentes e Inovação – FIWARE"

The workshop as held in December 17, 2014. Its main objective was to put together some of the main stakeholders from São Paulo about Smart Cities in order to gather information and experiences in order to foster new ideas and use cases. One the main concerns of all the participants is the technological growth of the Future Internet. During the discussions that took place in the workshop, the participants agreed that is necessary to explore the academic and economic opportunities offered by the Internet of Things. Thus, a common approach would be the creation of a stakeholders' community, which would include all the components of the innovation ecosystem, thereby providing lager integration amongst the human resources development.

In this event, we have the contribution of the following speakers:

- Prof. Dr. Sergio Takeo, LSI-TEC Director of Technology with a presentation "Brazilian Regional Centers For Future Internet Innovation"
- Eng. Jefferson Marcondes, member of the System Evaluation Group for Smart Cities debating about the "Background/Vision of the IoT Forum and the Perception on Smart Cities".
- Prof. Sérgio Risola, Executive Director of the Center of Innovation, Entrepreneurship and Technology(CIETEC) with the presentation "Contributions for the Smart Cities"
- Mr. Marrey Peres, Telecommunications Engineer with the presentation "Telecomunicações e as Cidades Inteligentes"
- Mrs. Cristina Cho, Active in the ICT area since 1999, with the presentation "Innovating at Brazil"
- Mrs. Maria Tereza Diniz, Executive Coordinator of the program USP Cidades, with the presentation "USP Cidades: Centro de Pesquisa em Cidades"

Another common approach during the workshop was the involvement of higher education institutions interacting with startups in the industry, along with incubators and accelerators. Furthermore, the Workshop counted on the involvement of funding agencies to enable the creation of financing channels related to smart cities. Another key point approached by the participants was the need to interoperability standards between different producers and developers.

During the discussions with the audience, some key-points of the debate with the presenters are as follows:

- Prof. Sergio Risola, reinforced the issue of the stakeholders with the statement: "results are the fundamental matter for you to achieve the trust of investors and companies".
- Mrs. Cristina Cho, reinforces that "the infrastructure in Brazil is deficient and expensive for small business; the lack of high qualified professionals make even worse this outlook for the small business and startups".

- Mr. Jefferson, answering questions of the audience about how startups can participate in the debate about standardization process, said that ABNT is qualified to handle such forum.
- Prof. Risola suggests the promotion of lectures to the incubated companies.

The most important result of the Workshop was the agreement of the audience and presenters on the creation of the Smart Cities stakeholder's community forum. This would enable a continuous discussion forum on Future Internet Technologies and its impact on Societal Challenges. One of the missions of this forum is foster innovations in Future Internet applications by SMES and Startups

Another important conclusion was about the financial support, which is the constant concern in the development of the Smart City application and/or project. In this case was agreed that USP and some participants of the audience would apply together for FAPESP funding at the programs PITE and PIPE, focusing solutions for Smart Cities.

Regarding to the results of the workshop, together with other USP representatives at the audience it was planned to create a Smart City tesbed inside several São Paulo University campuses based of the SmartCampus proof of concept presented at Section 3.1.

During the workshop discussions, security and standardization were considered as two key issues for the Smart Cities context. Thus, as result from the workshop, further discussion were planned with the audience in next similar events.

The workshop produced new perspectives about FIWARE in Brazil and Section 5.1 explore this other actions

4.2 Workshop “Inovação e Oportunidades com a Internet do Futuro”

After all the activities, detailed in Section 2 and Section 3 of this document, the workshop played a key role at FIWARE Regional Center deployment. The workshop had two important objectives: the first one was to enlarge the engagement of the local innovation ecosystem with FIWARE platform and the other one was to set up further actions at Uberlândia related to this ecosystem.

The workshop was held on December 16, 2014 at Uberlândia, MG. The event had the presence of several entrepreneurs and developers from different companies (SMEs). In addition, the public sector was present with a representative from the city hall and representatives from a telecom operator based at Uberlândia. The event put together different stakeholders from our local innovation ecosystem. The event happened outside the university, at I9UBERLANDIA site, with the involvement of SEBRAE and I9NAGI. About 30 different companies attended the workshop and had the opportunity to be introduced to the FIWARE platform, its architecture based on GEs and its infrastructure based articles in FIWARE Lab.

During the workshop, the FIWARE project was presented. The FIWARE Lab, the FIWARE software architecture and the reasoning behind them was explained to the audience. Finally, the proof of concept, detailed in Section 3.2, was demonstrated.

After the presentation, several stakeholders from the local innovation ecosystem could make their considerations regarding FIWARE opportunity that was presented. The debate was fruitful and could spark some next actions to be taken in the near future between UFU and those stakeholders.

One action was to do a new workshop, similar to the one that is being detailed in this Section, during January 2015. However, to this new workshop audience would be only startup companies from Uberlândia. This decision was made upon the request of some participants of the audience during the event.

The participants also asked for new presentations focusing specific areas of the FIWARE platform. These presentations would offer technical details about the FIWARE components and would provide use cases about their use. This request of the participants demonstrated their interest about the FIWARE platform.

Finally, some participants requested to have specific meeting with UFU in order to have an individualized support about the FIWARE platform. Section 5.2 explores in more detail some perspectives created after the workshop.

4.3 Training Post-graduation Students at USP

This course “Select Topics in Cloud Computing” (code PSI5120 2014.3 - Tópicos em Computação em Nuvem) is annually offered by the Electronic Engineering Postgraduate Department of EPUSP. The last edition occurred during the last trimester - from September 9, 2014 through December 12, 2014. The course was conducted by Prof. Sergio Takeo and with assistance of PhD candidates under his supervision. The goal of this 120-hour activity was to disseminate the FIWARE platform and resources using this technology as infrastructure to theoretical and hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

During twelve weeks, theoretical themes were covered, as well as a broad range of Cloud vendor platforms including AWS, Google App Engine, Microsoft Azure, Eucalyptus, and OpenStack. Eventually, professionals from those important market players – such as EMC and Microsoft - talked to students about resources and solutions of their companies.

In the 2014 version of this course, FIWARE was chosen as the infrastructure platforms for students develop and deploy their practical works. The amount of fourteen students enrolled in this course proposed a range of different Cloud applications using and integrating the FIWARE resources. Subjects addressed by students on their final work, all related to Cloud Computing, included crowdsourcing and crowd sensing, security, big data processing, work balancing, and legal aspects. Specifically, we can outline the results from two works:

- Development of an application to monitor air pollution and climate conditions using crowdsourcing and crowd sensing: a group of students developed a smartphone application based on open data sources and data collected from sensors, such as temperature, noise, humidity and other variables. All information gathered by the smartphone application was sent to the FIWARE platform to be store, processes and analyze, in order to produce different predictions, for example;

- Development of a GE to reinforce the characteristics of security of the FIWARE platform: another team developed a GE that implements authentication in the FIWARE cloud environment. This solution is based on Lyra2, a password-hashing scheme (PHS) with promising results in terms of security and performance.

.

4.4 Training Undergraduate Students at UFU

At the Faculty of Computing at UFU, there is a discipline called Design and Development of Information Systems #1 (GSI034) (Flavio de Oliveira Silva, 2014). This discipline is offered to undergraduate students that are almost concluding their studies and are close to go the market as professionals. The main goal of this discipline is to identify real problems, design, and develop software solutions in a standardized and efficient way. The discipline, under the responsibility of Prof. Flávio de Oliveira Silva, was offered from September 29, 2014 to February 23, 2015 to twenty two (22) students divided into four different groups with up to six participants.

During the last edition, that is still happening, one of the non-functional requirements that the software developed should comply was to use FIWARE GEs as part of the solution.

To accomplish this is was necessary to produce a guide called “FIWARE Based Application Development” (Flávio de Oliveira Silva, 2014) that was used by the students as a start point to understand FIWARE reasoning and the software architecture based on GEs. The guide also pointed to several resources available at the FIWARE Academy (FIWARE, 2014a). In order to understand FIWARE capabilities, the students conduct a research in order to retrieve technical information about the other applications created by using FIWARE GEs. Besides that, the students performed a technical study of FIWARE GEs and experimented them by using FIWARE LAB (FIWARE, 2014b).

The four groups of students selected different problems and proposed four different FIWARE enabled applications.

4.5 Participation at the Workshop “SwitchOn”

From January 8, 2015 to January 9, 2015, happened at Miami, Florida, USA, the “SwitchOn Workshop” (Liu et al., 2015). The Florida International University (FIU) hosted the workshop. The workshop was funded by the National Science Foundation (NSF) that also participate in the program with the two keynotes by its directors. The goal of this workshop is to search for collaboration between Brazil and USA regarding Future Internet Architectures.

UFU received an invitation to take part on this event and present some of the ongoing research. This was an opportunity to show the FIWARE approach in order to propose new services and applications focused on the Future Internet. Besides that, it was possible to present to several researchers, from different universities in Brazil, that work in the research area of Future Internet the FIWARE platform and FIWARE Regional Center concept that was deployed by USP and UFU.

The workshop was an excellent opportunity to get in touch with the right stakeholders that are working to build up the Future Internet in Brazil and in the USA. Moreover, the Brazilian participants was stakeholder that might host new FIWARE nodes in Brazil in the future.

During the workshop, common goals were identified between FIWARE and the effort conducted by NSF, called Future Internet Architectures (FIA) (NSF, 2014a) and FIWARE. NSF FIA is starting a new program, called NSF Future Internet Architectures (FIA) Next Phase (FIA NP) (NSF, 2014b) that aim to develop new services and applications based on new network architectures. Their focus is to create prototype systems, based on innovative solutions in order to leverage the existing Future Internet Architectures currently deployed. It possible to conclude that the focus now is bring the Future Internet results close to the market, focusing the innovation that Future Internet can enable.

Considering this context, the presentation of the FIWARE Regional Center concept created an interesting impact to audience because it demonstrated an action that is beyond of what USA is doing at this moment that is being handled by Europe and Brazil.

Finally, it is important to mention that another important result of this participation was that some researchers from different Brazilian universities get interested to use the FIWARE platform available in each Regional Center deployed.

4.6 Public Transportation Open Data

As a result, after the demonstration of the application, detailed in Section 3.2, we got the commitment of the Traffic Office from the municipality of Uberlândia to the activities of the FIWARE Regional Center at Uberlândia in order to provide the public data from all the buses of Uberlandia.

That data is not currently available, as the service providers are updating the equipment that is installed on each bus. As soon as they conclude this update, the data will be collected and transferred to the application. The software created to collect the public data considers a common interface that was agreed with the Traffic Office and this will enable an easier integration.

4.7 City of Uberlândia - Meeting with Mayor Representative

Another result related with the municipality of Uberlândia was a meeting, held on December 18, 2014, at the mayor office with its representative. During this meeting, the activities conducted with the Traffic Office were presented and this opened a channel to talk with other different offices from the municipality such as health and education. The basic idea is also gather public open data and store them at FIWARE Lab, fostering new types of applications in a context of a smart city.

This is an ongoing work and other meetings are planned to happen during 2015. Our approach in this case is to wait for the deployment of the **MyBus** Application the municipality of Uberlândia.

4.8 Scientific Paper Preparation

Preparation of a scientific paper with the title "Deploying Future Internet Applications by Using the FIWARE Approach". The authoring and preparation is a joint work between USP and UFU.

The paper describes an architectural pattern based on FIWARE GEs and the reasoning necessary to build services and applications based on the FIWARE approach. This contribution to the scientific community is the result of the teamwork conducted during by USP and UFU during the deployment of the FIWARE Regional Center. The contribution is the result of the reasoning that had to be developed in order to present and disseminate FIWARE to different stakeholders such as students, SMEs, software developers and entrepreneurs.

This work will be submitted to the 10th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom 2015) by the end of February 2015.

5 Perspectives

The work conducted by USP and UFU opened new perspectives at each Regional Center. In this section, we highlight some future perspectives base on the work conducted and presented in the previous Sections.

5.1 Regional Center at São Paulo (USP)

The activities conducted at São Paulo allowed a connection with new and important stakeholders such as CIETEC (CIETEC, 2015), USP CIDADES (USPCIDADES, 2015) and Brazilian IoT Forum (IOT FORUM, 2015). This connection may foster new FIWARE adopters in Brazil, especially SMEs and the public sector.

The leadership of USP can also bring together research funding agencies, such as FAPESP and innovation funding agencies such as FINEP and accelerators.

Considering the reach of work, USP will provide continuous information, support and training to local innovation ecosystem.

USP also plan to enhance FIWARE LAB node. The goal is the federation with FIWARE Lab at Europe. Also, to add more resources to the physical infrastructure considering aspects of security, availability and performance. Finally, we expect to offer new services based on the demands of the local ecosystem.

Considering previous expertise of USP on handling events with the community, we plan to continue workshops based on specific subjects such as “Smart Cities Water Management”, “Security at Smart Cities”, and “Managing Crowds during huge events”. These focused events will be linked with FIWARE platform and the participants will be directly related to theme. Each event will have the participation of selected SMEs that may have interest in each specific area and in this case will be potential adopters of the FIWARE approach do build innovative solutions.

These combined actions can contribute to create a loop were research, development and innovation based on FIWARE platform, conducted by SMES and entrepreneurs and by the academia contributing to bring the Future Internet to Brazil

5.2 Regional Center at Uberlândia (UFU)

At Uberlândia, considering a previous and strong connection from UFU with several companies and organizations that are part of the local innovation ecosystem, some interesting perspectives were created regarding the FIWARE Regional Center.

The application MyBus, presented at Section 3.2 will be updated and one of its new features is an optimized user interface for visually impaired. The Android app will be offered to all citizen of Uberlândia (about 250K daily users of the public transportation) with the support of the Traffic Office. We foresee this result is an important demonstration of FIWARE capabilities and an interesting case that will inspire other cities in Brazil.

A direct result from the Workshop, presented in Section 4.2 is a scheduled meeting that will happen in January 2015 with a local ICT company from Uberlândia called CEDRO (www.cedrotech.com) this company has about 100 employees and work with software products to the A market. CEDRO is interested to get more information about the FIWARE architecture in order to incorporate the FIWARE GEs on a new product that if focused on smart living based on different events that are configured by the end users. In this case, SEBRAE from Uberlandia was responsible for the connection with Company. We except that new meeting such as this one may happen in the next year with new companies. We are ready to help and support other companies in order to help them to incorporate FIWARE architecture at their products and services.

Another as a direct result from this first workshop, detailed in Section 4.2, is a new workshop, similar to the first one. This event is planned to happen at January 29, 2015, however, the invited attendees are Startups from Uberlândia. In the same way, new workshops may be set in a joint effort with the local innovation ecosystem in order to highlight specific aspects of the FIWARE platform.

Considering FIWARE dissemination by using Campus Party events, UFU is ready to help FIWARE at Campus Party Brazil 2015 in order to support on several evangelism activates such as workshops, lectures and training. This support by Brazilian institutions may improve the reach of FIWARE during Campus Party.

The open data from the Traffic Office of the municipality of Uberlândia will create new opportunities regarding services and applications that may explored by the local innovation ecosystem and by the ICT companies based at Uberlândia. New sets of data that may be available as long as the municipality provider other types of data will improve the capacity of the emerging of new other opportunities and possibilities.

In our vision, this has the power to create around the Regional Center a set of smart city based applications. This also can contribute to the sustainability of the FIWARE Regional Center at Uberlândia enabling its improvement regarding infrastructure, capacity and enabling new Future Internet based research with a tight connection with innovation.

6 Conclusion

The task, "Creation and Support of FIWARE Regional Centers in Brazil", produced several outcomes. After putting together all the results produced by USP and UFU, detailed along this report it is possible to assert that it reach its goals.

Moreover we can also state that this work, conducted during five months, will be capable to produce several results in Brazil relate to FIWARE from now on. Some actions were just the start point that will enable other results and outcomes, thus, having the capacity to enable a virtuous process regarding FIWARE at Brazil.

The work conducted reached a select group of stakeholders represented by SMEs, ICT related companies, the public sector, entrepreneurs and innovation related organizations at Uberlândia and São Paulo. The training actions started, and that will continue, will produce human resources that will be ready to contribute with these stakeholders, then increasing research, development and innovation related with Future Internet.

In each locality, the FIWARE Regional Center is becoming a reference about FIWARE and the Future Internet and our next actions will contribute to this goal. To do so, we would like to highlight some future work that we are capable and willing to conduct here in Brazil.

We plan to integrate other cities in Brazil, such as "Águas de São Pedro", for example, and other two cities integrated in the Brazilian Digital Cities Program. The selection of these cities will use different criteria such as touristic places, crowds sensing capabilities and available infrastructure. These criteria also will take into account the opportunities provided by funding available.

Regarding funding, we will work closely with funding agencies such as FAPESP, from São Paulo state and FAPEMIG, from Minas Gerais state. SEBRAE can also be a partner of this work. The goal here is to foster FIWARE base application development.

We also are planning to enhance FIWARE Lab nodes at each Future Internet Regional (São Paulo and Uberlândia) regarding physical infrastructure considering security, availability and performance and considering offers based on the demands of each local ecosystem.

Moreover, we are committed to contribute to the deployment of other FIWARE LAB nodes in Brazil offering support to the deployment and operation.

An important support action that we will focus on the next year is to provide continuous information, support and training to local innovation ecosystem by using local created content focusing the context of each region. To improve the offering of new open data, especially from the public sector in areas such as traffic and transportation, health, public security will also be a key action in each locality.

The FIWARE Regional Center at Uberlândia and São Paulo accomplished an important step during 2014. The actions and outcomes give us the directions to continue this work and the path to continue in order to find new results, based on FIWARE platform, which will collaborate to the deployment of Future Internet in Brazil.

References

CIETEC, 2015. Cietec [WWW Document]. URL http://www.cietec.org.br/?lang=en (accessed 1.10.15).

FIWARE, 2014a. FIWARE Academy [WWW Document]. URL http://edu.fiware.org/ (accessed 1.10.15).

FIWARE, 2014b. FIWARE Lab [WWW Document]. URL https://account.lab.fiware.org/ (accessed 1.10.15).

IOT FORUM, 2015. IoT Brasil Fórum de competitividade [WWW Document]. URL http://www.iotbrasil.com.br/new/en/ (accessed 1.10.15).

Liu, J., Ibarra, J., Alvarez, H., 2015. SwitchOn Workshop [WWW Document]. URL http://switchon.ampath.net/?page_id=12 (accessed 1.10.15).

NSF, 2014a. NSF Future Internet Architecture Project [WWW Document]. URL http://www.nets-fia.net/ (accessed 1.10.15).

NSF, 2014b. Funding - Future Internet Architectures - Next Phase [WWW Document]. URL http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504882 (accessed 1.15.15).

Silva, F. de O., 2014. GSI034 - Design and Development of Information Systems #1 [WWW Document]. URL http://www.facom.ufu.br/~flavio/pdsi (accessed 1.10.15).

Silva, F. de O., 2014. FIWARE Based Application Development [WWW Document]. URL http://www.facom.ufu.br/~flavio/pds1/files/2014-02/FIWARE%20Based%20Application%20Development.pdf (accessed 1.10.15).

USPCIDADES, 2015. USP Cidades [WWW Document]. URL http://www.uspcidades.net/en/home (accessed 1.10.15).