

A solution for tracking visitors in Smart Shopping environments: A real platform implementation based on Raspberry Pi

Juan R. Santana¹, Juan Carrasco², Jose A. Galache¹,
Luis Sanchez¹ and Ramón Agüero¹

Universidad de Cantabria, Santander, Spain

¹ {jrsantana, jgalache, lsanchez, ramon}@tlmat.unican.es

² jca37@alumnos.unican.es

Abstract. In parallel to the explosion of the use of wireless technologies to connect devices, the scientific community is continually aiming to take advantage of such technologies to provide new services. In this sense, there have been many attempts to exploit the information provided by IEEE802.11 and Bluetooth interfaces, commonly found in most of the smartphones that are being used at the time of writing. In this paper we describe a novel deployment that fosters such approach. Furthermore, the measurements that are gathered are made available, thanks to its integration within the SmartSantander testbed, and to the federation with complementary testbeds. The federation platform and the described deployment are outcomes of the FESTIVAL collaborative project (Europe-Japan). Besides depicting the corresponding software architecture, the paper also discusses some preliminary results that are used to assess the feasibility of the proposed scheme.

Keywords: IEEE802.11 · indoor localisation · federation · testbed · experimentation facility · EaaS · SmartSantander · smart shopping.

1 Introduction

During the last decade, the penetration of smartphones in the global market has strongly increased, providing a larger number of features for the users. Among all these features, connectivity is seen as one of the most relevant ones. Hence, most of the smartphones that are nowadays sold include wireless technologies such as 802.11 or Bluetooth. Furthermore, according to some usage statistics, three-quarters of smartphones users employ location-based services [1]. Although location technologies such as GPS or Glonass are well established, they cannot provide localisation information in indoor scenarios. Nowadays, there are many alternatives for indoor localisation that leverage existing Wi-Fi, FM, TV and GSM signals sent by the surrounding devices to localise themselves; as well as RF-beacons, RFID, infrared, ultrasound, Bluetooth, short-range FM transmitters or magnetic signal modulators to perform indoor localisation with a wide range of results [2].

Despite these multiple attempts that have been carried out for providing indoor positioning, there are no existing solutions yielding accurate results. This paper describes a novel testbed that would allow experimentation of radio technologies, such as 802.11 or Bluetooth, for indoor localisation in a real scenario. The approach followed in this infrastructure is to deploy a set of SSH-accessible devices with the appropriate sensors, including IEEE802.11 and Bluetooth 4.0 interfaces, as well as complementary sensors, such as humidity and temperature. Its flexible design would allow the integration of more sensors in the future.

The infrastructure deployed for experimentation has been built under the framework of the FESTIVAL project [3], a European and Japanese collaborative project that aims provide a federation platform for external experimenters, where the focus is on three domains: Smart Energy, Smart Building and Smart Shopping [4]. In this regard, within the framework of the Smart Shopping domain, the infrastructure is integrated within the SmartSantander testbed [10], which will be federated in the EaaS FESTIVAL platform. Hence, the deployment follows a two-fold approach. On the one hand, to provide external experimenters with a set of measurements to perform experiments to improve indoor localisation techniques based on radio technologies. On the other hand, the implementation of a localisation service that can be exploited by end-users.

The paper is structured as follows. Firstly, Section II presents the FESTIVAL initiative for the creation of an intercontinental federation platform, namely EaaS (Experimentation as a Service). This section also includes the description of the SmartSantander testbed, that will be federated in FESTIVAL. Secondly, Section III describes the infrastructure deployment, including the different sensors that have been installed and the localisation of the devices in the deployment area. Afterwards, Section IV describes the system design, conceived to retrieve the measurements, as well as its integration within the SmartSantander platform at the server side. Section V discusses a first analysis of the measurements gathered from the deployment, as an assessment of the feasibility of the platform for experimentation. Finally, Section VI presents the conclusions derived from the work carried out, as well as the next steps to be performed in the next years of the FESTIVAL project.

2 FESTIVAL intercontinental federation and SmartSantander

As described in the introduction, one of the main goals of FESTIVAL is to provide an EaaS platform for external experimenters that will federate a set of facilities from Europe and Japan. The EaaS platform will provide a single API to access a highly heterogeneous set of resources. This API will enable the rapid deployment of experiments based on the required services in some of the most demanded domains for experimentation: smart energy, smart building and smart shopping. Moreover, the EaaS platform provides easy repeatability and traceability of experiments in different testbeds, as well as resources for conducting any experiment. The facilities federated in FESTIVAL can be divided in four groups, depending on the resource type:

- **IT testbeds**, which embrace those facilities that provide virtual resources for computation. These cope with the computing power requirements of applications for smart services.

- **Open Data platforms**, providing a SPARQL query languages to perform semantic searches in heterogeneous smart city platform on a uniform way. These platforms provide large datasets from several cities, such as Lyon or Santander.
- **IoT testbeds**, these are related with the Internet of Things domain, where multiple sensors and actuators are made available to the experimenter. The data gathered from these sensors is provided in real time, and there are many of them from the smart city, industry or building automation domains.
- **Living labs**, which are composed of open spaces where the experimenters can include prototypes or gather the opinion from the end users. For instance, they can take advantage of the living labs to perform surveys or focus groups about the developed applications or services.

Following the aforementioned division, the federation is composed by four gateways, which manage the testbeds. The gateway in charge of IT resources is based on SFA (Slice based Federation Architecture) [5], as it is already used in other FIRE projects such as FED4Fire [6]. The IoT gateway is implemented through the sensiNact platform [7], while the Open Data and Living Lab gateways are ad-hoc developments. Furthermore, on top of the gateways, FESTIVAL implements a logic for provisioning the EaaS RESTful APIs, the security layer, implemented using the IdM Generic Enabler from FIWARE [8], and the KPI (Key Performance Indicators) monitoring. Finally, a web portal is offered to the experimenters, to ease the experimentation. Fig 1 shows the logic architecture of FESTIVAL.

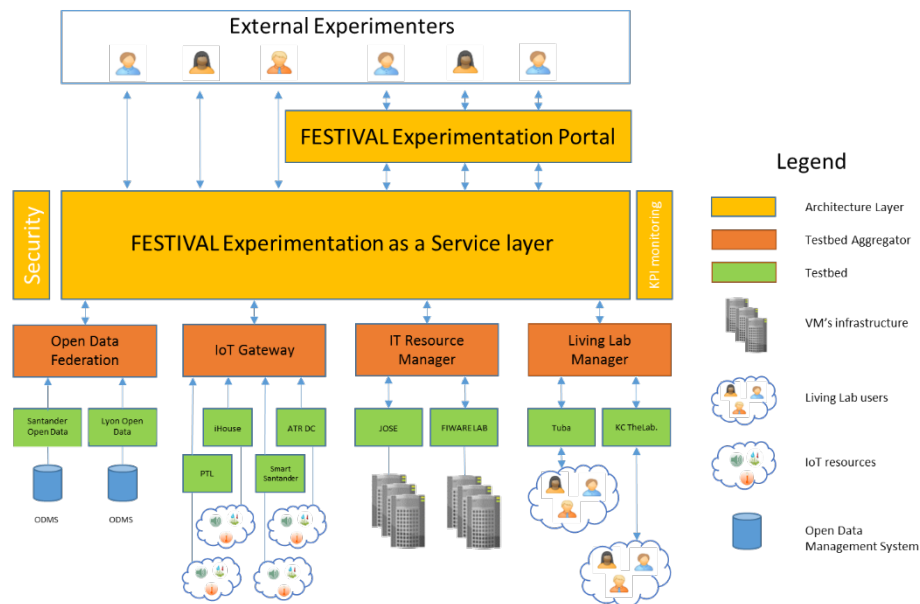


Fig 1. EaaS FESTIVAL architecture overview

Among others, one of the testbeds that is being federated in the EaaS is the SmartSantander testbed [9] [10]. SmartSantander was conceived as an urban laboratory, which provides a unique infrastructure to experiment with the Internet of Things sensors, such as traffic, environmental or mobile sensors, and their communication protocols, including exclusive native IEEE802.15.4 interfaces for experimenting purposes. Additionally, SmartSantander also envisions the provision of services for the citizen, easing their daily life.

The SmartSantander's deployment size involves more than 12000 IoT devices, which can be divided, depending on the sensor type, as follows:

- **Static environmental monitoring:** these devices are the core of the SmartSantander testbed, and are composed by around 2000 IoT devices with several sensors, including: temperature, luminosity, CO or noise sensors.
- **Mobile environmental monitoring:** deployed to extend the static sensors, these nodes have been deployed in around 150 public vehicles, such as buses and taxis, able to retrieve information from the CO, NO₂, O₃, particle matters, temperature and humidity sensors included in each module. Other driving parameters are also gathered from 10 of them, including position, altitude, speed, course or the odometer.
- **Parks and gardens irrigation:** with 50 nodes deployed in three parks in Santander, these sensors are able to measure parameters such as temperature, humidity, ground temperature or soil moisture tension. All of them are irrigation-related parameters aim at improve its efficiency.
- **Outdoor parking management:** buried under the asphalt, almost 375 sensors are deployed in the city centre of the city to monitor existing parking spots.
- **Guidance to free parking lots:** deployed to provide the parking information to the drivers, 10 panels are available in some streets of the city centre, indicating the free spots in the area.
- **Traffic density monitoring:** are 60 sensors deployed in the main entrances of the city to measure the traffic parameters, including traffic volumes, road occupancy and the average and median speed of the vehicles.

In this sense, the infrastructure that is described in this paper can be seen as the first indoor deployment in the SmartSantander testbed.

3 Infrastructure deployment

The chosen location to deploy the system and gather data about visitors is a well-known market in the city centre of Santander. The market is an old building that was restored in 2000 to make room for shops, restaurants, a regional tourist office and a museum. The market is called "Mercado del Este" and is a symmetric building of 60x40m, with three entrance doors at each long side. The interior of the market is composed by a two aisles that cross the market side to side, and one long corridor surrounding its interior. The market map is represented in Fig 2, which also highlights the location of the deployed devices, including its coordinates referenced in meters to the lower left corner.

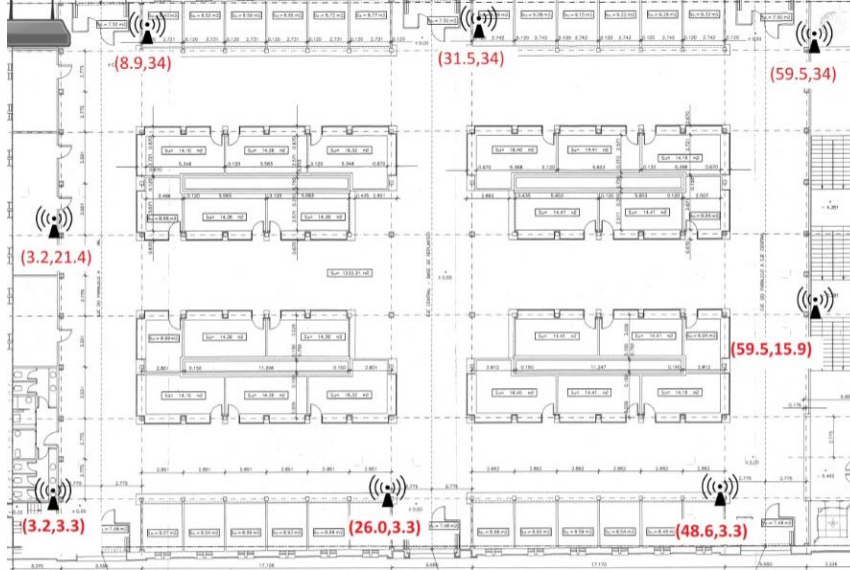


Fig 2. Deployment site and the location of the measurement nodes.

The market location is optimal since, being at the downtown, lot of people visit it every day, guaranteeing enough measurements for future experiments. It counts with access to the municipality network, which provides 24h of high speed connectivity. Finally, within the market, it also counts with a gateway from the SmartSantander network, providing connectivity through Digimesh, a proprietary protocol based on IEEE802.15.4. Hence, SmartSantander compatible repeaters can be installed.

To provide a great set of variable measurements, we have deployed 8 devices, configured for the IEEE802.11 based localisation.

The hardware platform chosen for the deployment is the Raspberry Pi. This is an ARM based platform that provides the required USB ports for the interfaces (IEEE802.11 and Bluetooth) plus a set of GPIO pins to support the extra environmental sensors. We have deployed 7 Raspberry Pi model 2 and 1 Raspberry Pi model 1 B+. The main technical specifications of the Raspberry Pi are described in Table 1.

Table 1. Technical specifications of the deployed devices.

Parameter	Raspberry Pi Model B+	Raspberry Pi 2 Model B
SoC	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM 1176JZF-S a 700 MHz	900 MHz quad-core ARM Cortex A7
GPU	Broadcom VideoCore IV	Broadcom VideoCore IV
SDRAM	512 MiB	1 GB

USB ports	4	4
Energy Consumption	3 W	4 W
Dimensions	85.60mm × 53.98mm	85.60mm × 53.98mm

The operative system used in both types of Raspberry Pi is a debian image named “Raspbian GNU/Linux 8 (jessie)”. This operative system, along with the well-known Raspberry Pi hardware, has been chosen mainly due to the wide community supporting it. Additionally, it might be easily extended in the future thanks to the existing USB ports.

For this first deployment, we have integrated 3 types of sensors: two of them are based on radio technologies (localisation purposes), while the third one is devoted to environmental monitoring, providing measurements of temperature and humidity.

The radio interfaces are two: a Bluetooth radio interface and an IEEE802.11n radio interface. The former one is a generic Bluetooth chip that supports Bluetooth 4.0, whilst the IEEE802.11 interface uses the TP-LINK TL-WN722N USB adaptor, with a 4 dBi antenna. This adaptor implements the Atheros AR9271 chipset, that is fully compatible with the selected linux distribution and provides the IEEE802.11n specification. Currently, the IEEE802.11n radio interface will be dedicated for experimenting purposes, while the Bluetooth interface would be later exploited to send real-time offers, being out of the scope of this paper.

Regarding the environmental monitoring, the sensor used is a DHT22, whose specifications are shown in Table 2, as described in its datasheet.

Table 2. Technical specifications of the deployed environmental sensors.

Parameter	DHT22
Power Supply	3.3-6 V DC
Output signal	Digital signal via single-bus
Sensing element	Polymer Capacitor
Operating Range	Humidity 0-100%RH and temperature -40°-80° Celsius
Accuracy	Humidity +-0.1%RH and temperature +-0.5° Celsius
Resolution	Humidity +-0.1%RH and temperature +-0.1° Celsius
Repeatability	Humidity +-0.1%RH and temperature +-0.2° Celsius
Dimensions (mm)	14 × 18 x 5

All the devices are powered 24 hours through Power Over Ethernet. This guarantees continuous data provision with configurable measurement rates.

4 Software platform for data gathering

As mentioned earlier, the infrastructure deployment is able to gather several types of measurements that will enable the experimentation in the EaaS platform within the Smart City domain. The sensor information elements that will be captured are described below.

Firstly, the location data for experimentation is obtained using the IEEE802.11 interface. The IEEE802.11 protocol defines the two lower layers of the OSI model, the physical and link layers, within the open frequency range in the 2.4GHz. At the time being, most common versions of the protocol are the IEEE802.11g and IEEE802.11n, supported by the deployed devices in the infrastructure, while the last version in the market is the IEEE802.11ac. It is worth mentioning that all versions are backwards compatible and it is possible to capture frames from devices implementing a more recent version.

The IEEE802.11 protocol specifies two methods to find already known access points. The first method is the so called “passive scanning”, in which the device is listening to Beacon messages that are periodically sent from routers. This is a high energy consuming method, as the devices must be continuously listening to beacons. On the contrary, the most common method is the “active search”, in which the device periodically sends a so-called “Probe Request” frame, with the known access points in all the 802.11 channels, and waits for a reply from one of them. Due to the fact that this method is used by the client devices, and the frames are periodically sent, we will capture these data packets in order to gather the required features: the RSSI and a unique identifier.

Secondly, 4 of the deployed devices also integrates the DHT22 sensor described in the previous section. These data will be also sent to the SmartSantander, to provide extra measurements for experimentation.

Finally, we will also deploy a dosimeter, which will provide information regarding the electrical field inside the building, which could be therefore correlated with the data got by the 802.11 interface.

4.1 Software controller for the measurement nodes

The software controller for data gathering has been implemented in python. The library used for parsing the 802.11 packets is pyshark [11], which takes advantage of the command-line utility of wireshark, named tshark. For the case of the environmental parameters, the measurement process has been made using the library provided by Adafruit, Adafruit_DHT [12].

The controller divides the gathering of sensor measurements into two processes, one for the environmental measurements, and another one to read the 802.11 probe request packets. Once the program is started, both processes are created. The process for measuring the environmental data is described below:

- 1) The sensor is powered. Afterwards, a loop is initialised to measure the data provided by the sensor every 60 seconds.

- 2) Within the started loop, every 60 seconds the sensor data is captured, by reading the GPIO pins where the sensor is connected, up to 5 times in a row. Then, we just consider the median of the five values.
- 3) The measurement is sent through a TCP socket to the server, where it will be injected into the SmartSantander testbed. The data message sent to the proxy includes: the temperature with a precision of two decimals, the humidity with a precision of two decimals, and the time when the measurement was taken.

On the other hand, Fig 3 shows the workflow of the software controller for the 802.11 radio interface, which is also described as follows.

- 1) Firstly, the process checks if the 802.11 interface has been initialised. If it is not the case, the interface is initialised with the monitor mode enabled. Using the monitor mode allow us to gather all the existing packets in the 802.11 wireless channels.
- 2) Once the interface is up, the process start listening to the packets in the wireless channels, filtering them in order to retrieve the Probe Request packets.
- 3) Every time a probe request is captured, the controller draws the main features of the packet for localisation and device counting: MAC address and the RSSI. So as to preserve the privacy of the users, the MAC is anonymised by means of a hashing function that implements SHA-1.
- 4) Every time the probe request packet received is from a new device, the controller creates a new field in a local cache, storing the MAC address and the reception time. Hence, it is possible to set up a minimum period of time for each packet sending.
- 5) Finally, the packet is sent to the server if there were not measurements sent during the last time period of 10 seconds (so as to get all the measurement nodes synchronised, all of them implements a network time protocol, and the periods are the result of dividing each minute in 6 periods of 10 seconds). The message is composed by the hashed MAC, the RSSI, the timestamp when the packet was received in the interface, and the timestamp when the measurement node sent the packet to the server.

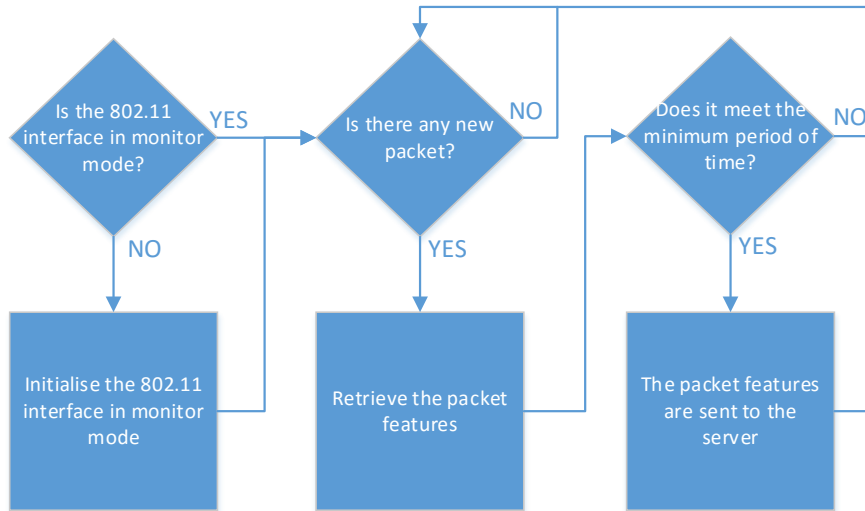


Fig 3. Workflow of the software controller for measuring the RSSI from IEEE802.11 packets.

4.2 Server software and SmartSantander integration

As was mentioned earlier, the measurement nodes deployed in the “Mercado del Este” have been integrated into the SmartSantander platform, so they will be part of the EaaS FESTIVAL platform. In order to forward the measurements gathered to the platform, we have deployed a central server to receive such measurements and work as a gateway. The aim of this server is twofold: on the one hand, to allow the storage of the measurements into an internal database during the testing phase; on the other hand, to connect the server software output to the existing SmartSantander interfaces to include the data as part of the testbed.

The communication between the deployed devices and the server has been performed using TCP sockets. Although at a first phase we used direct communication against a temporary database, the number of measurements gathered per second was delaying the file storage process. Hence, to avoid such situation, we used lightweight TCP sockets to send the data from the measurement nodes to the server. The management of the socket was performed using the ZeroMQ libraries, that are able to manage up to 500000 messages per second with a latency of 100 μ S [13]. ZeroMQ is a networking library that uses sockets to send atomic messages, providing several types of communication patterns, such as PUB/SUB, the sender (PUB) characterises the messages and any number of receivers (SUB) can subscribe to them, or PUSH/PULL, where the sender (PUSH) sends a message to a specific address created by the receiver (PULL). The communication diagram between the deployed devices and the server is depicted in Fig 4.

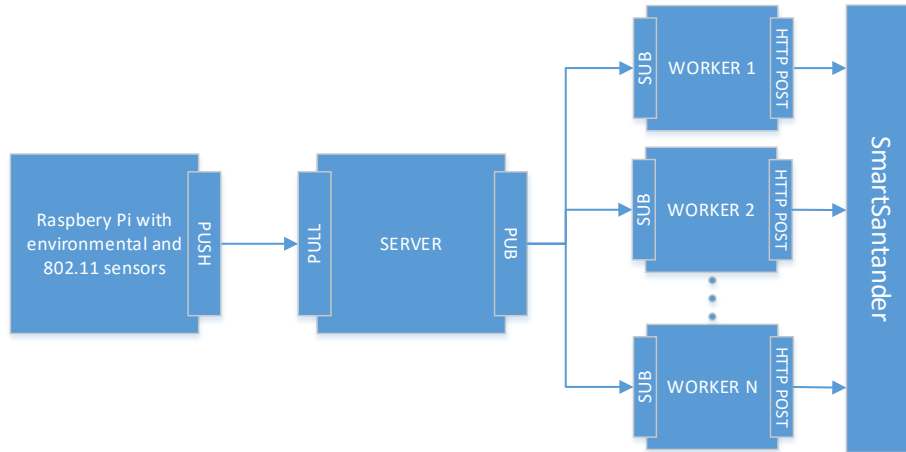


Fig 4. Communication diagram of the deployed nodes and the server.

- 1) The first step is to create the socket in both sides. In the server side, a PULL socket is bound to a port, and it start listening to a configured port for incoming messages. In the measurement nodes, a connection with the IP and the port of the server is performed through a PUSH socket. Additionally, the server initialises a local PUB socket where it will publish all the incoming messages. At the execution time, the server will also create eight workers that will be subscribed to the PUB socket.
- 2) Secondly, the measurement nodes will send a message through the PUSH socket including the parameters mentioned in the software controller description.
- 3) The server that is listening for the incoming messages will forward every message to the PUB socket.
- 4) Finally, the workers subscribed to the SUB socket will forward every message to the SmartSantander testbed with the appropriate message format. The interface used in SmartSantander is a web service interface, and we consider it as a black box.

5 Deployment validation

We have performed several tests to validate the deployment carried out and confirm its appropriate behaviour. These tests can be divided in two different parts. Firstly, there is a comparison between the temperature, the number of people in the deployment site and the number of detected devices by the system. Secondly, the comparison is done between a set of known positions of a specific device and the estimated positions using a weighted centroid method.

5.1 Comparison between the devices detected and the environmental measurements

The first comparison has been done between the detected devices using the IEEE802.11 interface and the environmental measurements. Additionally, we have also performed real measurements, by counting the number of people in the deployment site during two hours, with a six minutes' period of cadence. This is intended to assess the correct behavior of both type of sensors, as well as to get a first impression of the capacity of the system to count people based on the number of detected devices.

Fig 5 and Fig 6 show the time series of the number of detected devices in the deployment site and the number of people present at that moment in the building. The number of the detected devices are considered obtaining the number of different hashed MACs detected during a period of 2 minutes, with a frequency of one measurement every 6 minutes.

considers the different received hashed MACs during the last 2 minutes. In order to avoid the detection of devices that are from people walking outside the building, we consider that a device is within the building when a minimum number of measurement nodes have detected it. In the Fig 5 we have considered that a device is within the building when, at least, 4 of the 8 IEEE802.11 sensors have indeed detected it. In the Fig 6 the minimum of measurement nodes considered are 6.

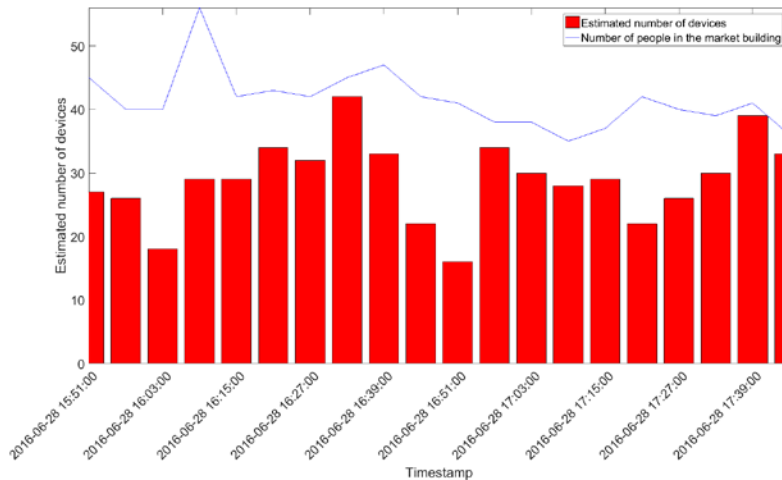


Fig 5. The number of people counted in the market building and the estimated number of devices detected by the deployment, considering the detection of 4 measurement nodes.

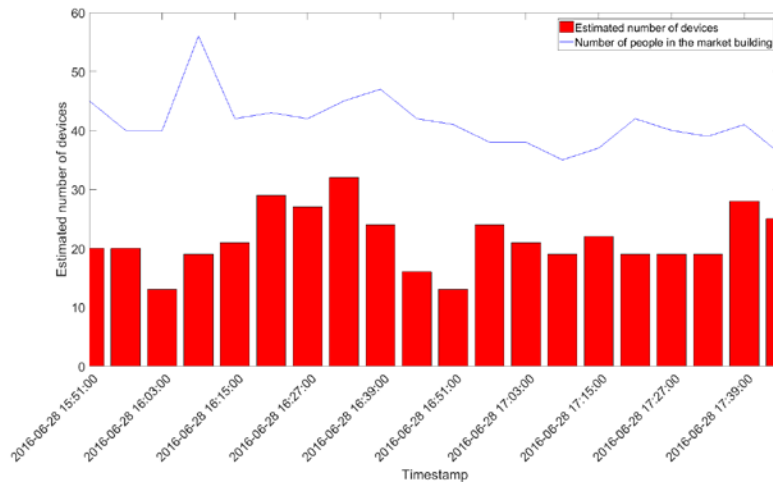


Fig 6. The number of people counted in the market building and the estimated number of devices detected by the deployment, considering the detection of 6 measurement nodes.

As can be seen in both figures, the number of detected devices is always smaller than the number of people in the building. This could be happening because there are yet some people within the market that are not using smartphones with Wi-Fi capability. It is also possible that some people present at the building had the Wi-Fi connectivity switched off. Furthermore, the cadence of the probe requests sent by the mobile phones heavily depends on each manufacturer, and it can last for more than 15 minutes. Therefore, detecting the devices strongly depends on the sojourn time of the people in the building.

On the other hand, Fig 7 and Fig 8 show the temperature measured by the different sensors and the number of detected devices during one day. These graphs are shown to assess the feasibility of the platform to measure and manage other type of parameters, such as environmental values. Therefore, to provide the possibility to carry out studies involving different domains which are not correlated a priori.

According to the figures, we can consider that the system is able to provide a good approximation of the occupancy state of the building, as the system detects the devices within the opening times of the building; and the hours with more visitors, lunch and dinner periods, match with the higher number of detected devices. Regarding to the environmental parameters, the correlation with the number of detected devices is different. Whilst the temperature seems to be higher when most of the devices are detected, the humidity seems to follows a different trend, being lower as more devices are detected. However, there are other factors that affect to the measurements of the environmental parameters and their relation with the number of people in the building, such as the air conditioners, ventilation or the external conditions, thus it is not possible to confirm the causality with the current data.

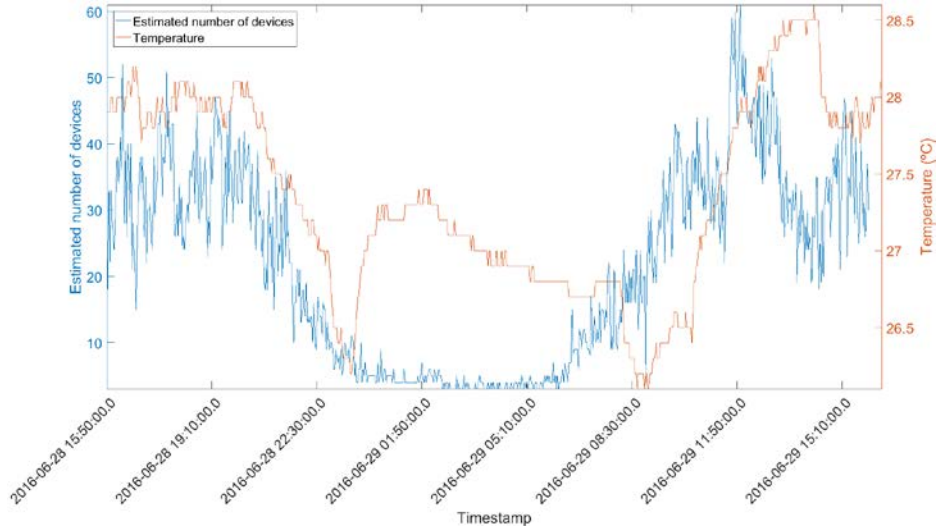


Fig 7. Comparison between the measured temperature and the detected devices during 24 hours.

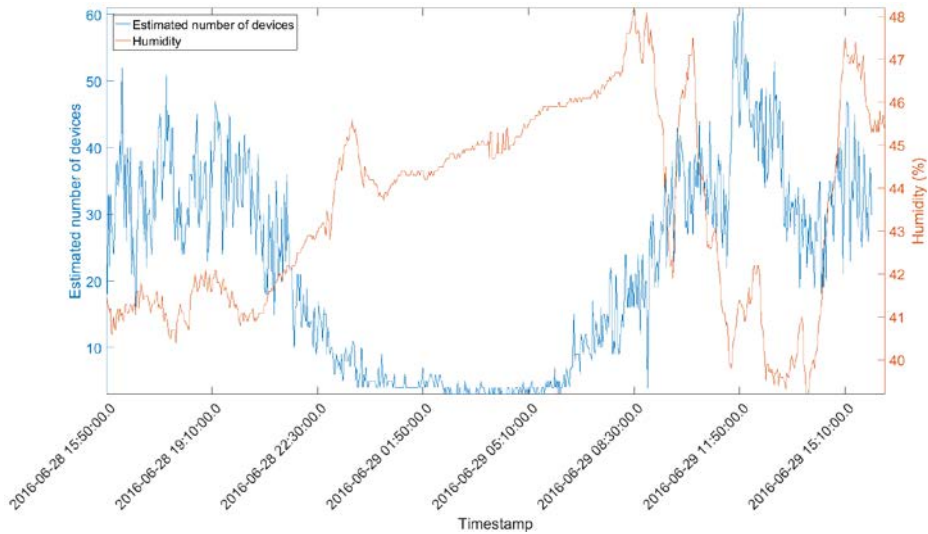


Fig 8. Comparison between the measured humidity and the detected devices during 24 hours.

5.2 Location estimation using a simple weighted centroid-based method

In order to test the behavior of the IEEE802.11 deployment for locating people, we have implemented a simple algorithm based on weighted centroids [14][15]. This

algorithm uses the RSSI to weight the coordinates of the measurement node that detects a device.

The centroid, or geometric centre, of a plane figure can be defined as the arithmetic mean position of all points in the shape. The calculation of the centroid can be done with (1).

$$(X_{target}, Y_{target}) = \left(\frac{1}{N} \sum_{k=0}^N X_k, \frac{1}{N} \sum_{k=0}^N Y_k \right) \quad (1)$$

Where X_k and Y_k are the coordinates of the deployed measurement nodes that detected the node K , being N the number of measurement nodes.

However, considering that the measuring devices are static, the centroid solution does not use the RSSI values and the results can be limited to a determined fixed position for each of the different combinations of measurement nodes, as it will consider only nodes that detect a signal from a device. Therefore, to get more accurate results, the coordinates of the measuring devices are weighted by the measured RSSI (2).

$$(X_{target}, Y_{target}) = \left(\frac{\sum_{k=0}^n X_k \cdot W_k}{\sum_{k=0}^n W_k}, \frac{\sum_{k=0}^n Y_k \cdot W_k}{\sum_{k=0}^n W_k} \right) \quad (2)$$

Where, considering the RSSI as proportional to the inverse square of the distance, W_k is the root of the RSSI detected by the node K .

In order to analyse the performance of the system, we have taken manually a set of 95 measurements for a given device in 13 known positions. Applying the Weighted Centroid formula to the gathered measurements, we obtained a root mean error square (3) of 5,7242 meters. Fig 9 shows the histogram with the repeated square errors, yielding a reasonable approximation.

$$RMSE = \sqrt{\frac{\sum_{k=0}^n (X'_k - X_k)^2 + (Y'_k - Y_k)^2}{n}} \quad (3)$$

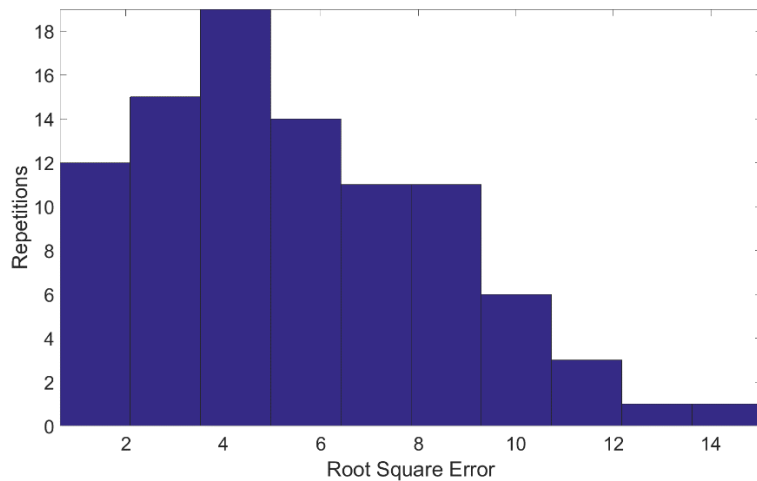


Fig 9. Distribution of the detected root square error.

Finally, Fig 10 depicts the real and the estimated positions of several measurements. It is clear that, in the case there are several measurements in a short period of time, we could avoid the results that are not grouped, which will reduce the variance error.

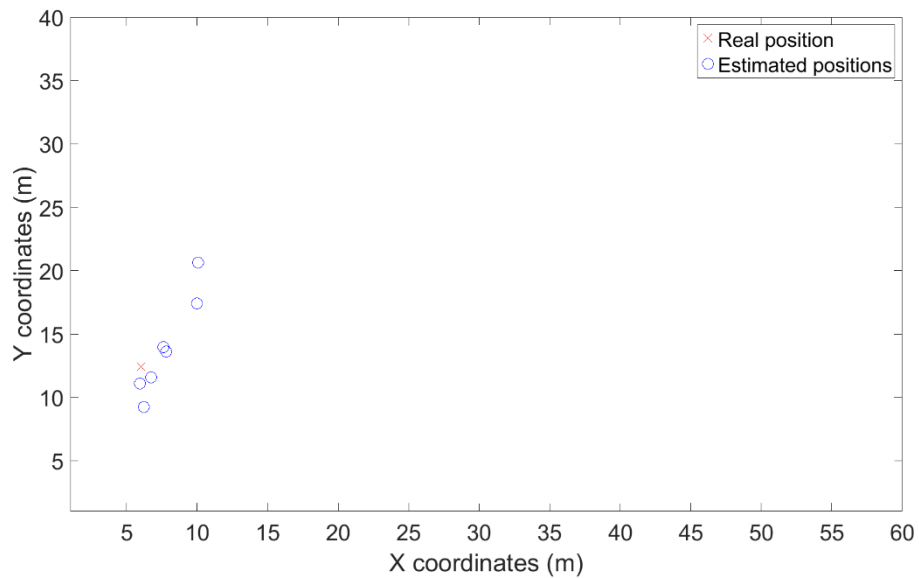


Fig 10. One measurement of a real position and estimated position using the weighted centroid-based method.

6 Conclusions and future steps

Online shopping has taken a big proportion of the traditional market consumers during the last years, considerably reducing sale numbers for local shops. In that sense, with the aim of modernizing these traditional local shops, many attempts have been done to provide new elements for attracting new generations of customers. Among them, localisation in indoor scenarios has been proposed in previous works, with a two-fold objective: on the one hand, to provide information to the customers, and on the other hand, to analyse their behaviour so that shop owners can take appropriate actions to improve the sales. These objectives have been addressed within the framework of FESTIVAL, through the deployment of an open real indoor area for experimentation, that is integrated in the SmartSantander testbed and will be accessible through the EaaS platform for the scientific community.

Aiming at providing a powerful tool to the scientific community to continue the research within the Smart Shopping domain, a testbed deployment in a real scenario has been presented. The deployment characteristics, including the devices used and the sensors included have been described in detail, as well as the deployment location and the software platform developed to sustain the system. Additionally, the FESTIVAL federation initiative has been presented, along with the description of the SmartSantander testbed, where the deployment has been integrated.

A first analysis of the measurements has been performed to assess the feasibility of the proposed testbed, including the analysis of real measurements and the comparison with the obtained results. We checked the accuracy of the occupancy estimation, as well as a localisation method based on weighted centroids.

Finally, as the deployment phase has just finished, the deployment access through the FESTIVAL federation will be released in the coming months, when the EaaS FESTIVAL platform is finished. Furthermore, new experiments are also envisioned, such as the implementation of new methods for localising people in the market, as well as improving the existing ones, such as the centroid-based method.

Acknowledgments. This work was funded in part by the European Union's Horizon 2020 Programme of the FESTIVAL project (Federated Interoperable Smart ICT Services Development and Testing Platforms) under grant agreement 643275, and from the Japanese National Institute of Information and Communications Technology. This work has been also supported by the Spanish Government (Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, FEDER) by means of the project ADVICE (TEC2015-71329-C2-1-R).

References

1. Zickuhr K.: Three-quarters of smartphone owners use location-based services, Pew Research Center's & American Life Project (2012).
2. Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., & Sen, S.: A Realistic and Comparison of Indoor Location Technologies: Experiences and Lessons Learned. In: Proceedings of the 14th international conference on information processing in sensor networks, pp. 178-189. ACM (2015).

3. FESTIVAL project, <http://www.festival-project.eu/>
4. Santana, J. R., Galache, J. A., Akiyama, T., Gurgun, L., Matsuoka, M., Maggio, M., & Murata, S.: FESTIVAL: Towards an intercontinental federation approach. In: International Conference on Mobile Networks and Management, pp. 269-280. Springer International Publishing. (2015).
5. Slice-based federation architecture.: Slice-based federation architecture (sfa). Ad Hoc Design Document. (2010).
6. FED4Fire project, <http://fed4fire.eu/>
7. SensiNact platform, <http://open-platforms.eu/library/sensinact-aka-butler-smart-gateway/>
8. FIWARE initiative, <https://www.fiware.org/>
9. SmartSantander project, <http://smartsantander.eu/>
10. Sanchez, L., Muñoz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., Pfisterer, D.: Smartsantander: IoT Experimentation over a Smart City Testbed. Computer Networks J., vol. 61, pp.217 -238 (2014)
11. Pyshark software, <https://github.com/KimiNewt/pyshark>.
12. Arduino DHT library, https://github.com/adafruit/Adafruit_Python_DHT
13. ZeroMQ Measuring messaging performance, <http://zeromq.org/whitepapers/measuring-performance>
14. Kosović, I. N., & Jaguš, T.: Enhanced Weighted Centroid Localization Algorithm for Indoor Environments. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol:8, No:7, pp. 1219-1223 (2014)
15. Dong, Q., & Xu, X.: A Novel Weighted Centroid Localization Algorithm Based on RSSI for an Outdoor Environment, Journal of Communications Vol. 9, No. 3, pp. 279-285 (2014)