Performance evaluation of Frameworks for iBeacon-based Localization using Bluetooth Low Energy

Paolo Bellavista, Antonio Corradi, Mattia Ferrari, Luca Foschini,
Pasquale Carlo Maiorano Picone, Rebecca Montanari
Dipartimento di Informatica – Scienza e Ingegneria
University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
{paolo.bellavista, antonio.corradi, luca.foschini, pasquale.maiorano4, rebecca.montanari}@unibo.it

***Abstract*—Indoor location is crucial for enabling the provisioning of novel location-based services in several areas. There are many different technologies that could be used in order to implement a reliable indoor location service, such as, among others, Beacon with Bluetooth Low Energy, Ultrasound wave system with special receivers. The choice between these technologies must take into account multiple parameters (e.g., user-centrality, proactivity,..), and should be driven by not only industrial relevance but also cost-effectiveness issues. In this paper we claim that Beacon with Bluetooth Low Energy is the best solution for indoor localization, and we present an extensive comparison between two frameworks that are the most relevant solutions interacting with beacon technology: Apple Core Location Framework and Estimote Monitoring. The comparison results offer a clear vision of which framework has to be used in order to implement a reliable and cost-effective indoor location-based service.**

Beacon protocol, indoor location, Bluetooth low energy, BLE

I. Introduction

Nowadays location-based services are commonly used for analyzing some aspects of user behavior, such as movements within an area, for emergency systems, and for simple proximity advertisement. Location-based services are differentiated in two main categories: outdoor location and indoor location. Outdoor location services can exploit a wide variety of technological support, such as GPS sensors, cellular networks, and public Wi-Fi hotspots. In order to determine a reliable position in terms of latitude and longitude (absolute position), which helps also to determine the relative location from a fixed point, all data acquired by sensors could be mixed in for obtaining a higher precision. For instance, we could use Wi-Fi signal strength with nearby hotspots mixed with GPS location and consider data from other sensors like accelerometers in order to determine the position and direction of a user.

On the contrary, for indoor location, there are fewer options in terms of sensors that can be exploited, and GPS and cellular data cannot be considered because they tend to be unreliable. There is the need to rely on different kinds of measurements based on different technologies and connectivity. For instance, Received Signal Strength Indication (RSSI) through Wi-Fi signal where there are fixed-position wifi access points which provide also connectivity for users; Ultrasound wave systems like ActiveBat [1], infrared like ActiveBadge [2] and RFID. However, some of these technologies have been mostly applied in experimental settings and seem to be not particularly adequate, because of the high number of required devices to provide indoor location and device heterogeneity.

Effective technologies for indoor location can be those that can handle the typical use case where the user does not have any network connectivity and is not willing to pay, in terms of battery consumption, the cost of an always active Wi-Fi sensor. Candidates are technologies such as Zigbee, Bluetooth, Ultrawide radio wave and many more. Zigbee is a very robust and reliable protocol in industrial ecosystem but has not a single appliance in consumer market; Ultra-wide radio wave cannot be used in heterogeneous environment because of the possibility of radio signals interference (like radio stations, emergency channels and many more); Bluetooth, instead, is a widespread technology in consumer market and Low Energy version is the best solution for multipurpose devices including smartphones.

The choice between these technologies must be made considering different parameters: the system must be user-centric, proactive and must be tailored to the particular context in order to be application-centric, and least but not last it must be industrial relevant. In [3] the differences between the first and second generation of Location Based Services (LBS) have been examined, concluding that second-generation LBS solutions are moving towards user-centricity, do not require any initial input from the user, and must be application centric. Taking into account also the business relevancy LBSs also must be cost-effective to allow their deployment also within startups or small/medium enterprises with a limited budget.

In order to accomplish suggested LBS requirements and have also an industrial relevancy, we claim that the use of Bluetooth low energy devices using RSSI is a good choice for indoor location. RSSI is one of the most popular methodologies for an indoor location in industrial applications because it achieves the aforementioned LBS requirements, particularly in terms of privacy preservation and user-centrality. The RSSI model suggests using a device with a fixed and predetermined position and the desired connectivity, in order to have fixed coordinates for distance calculation from our source.

In this work, we decided to choose Apple iBeacon with Bluetooth Low Energy because it has a very low error rate also on worst conditions as it stems from [4]. As a key contribution of this work, we have compared Apple's core location framework capable of interacting with iBeacon Bluetooth Low Energy and Estimote monitoring in order to determine which is more accurate and reliable in small rooms where signals might be overlapped and there is a lot of interference in presence of crowd. This comparison was made in collaboration with My Voice, a small/medium enterprise (SME) located in Emilia Romagna which is interested in provisioning novel indoor location and mobile proximity advertisement services. SMEs are increasing their interest in indoor location because it allows proximity marketing and various internal location-dependent applications, such as opera explanations within a museum pushed to the user device when it is near the target opera. We believe that this comparison is really important for two main reasons. First of all, it is the first comparison in terms of frameworks, secondly, results are relevant for guiding a business-oriented project. For instance, our results could help deciding whether to use or not an RSSI based indoor location service using Estimote Monitoring within a museum to track museum's visitors and to send them a detailed explanation when they are in front of an opera.

The paper is organized into three main sections: background, experiment design and deployment description and experimental results. In the background section, we will describe Bluetooth Low Energy, Beacon protocol, and introduce related works made with Beacon Bluetooth Low Energy. The third section explains how we designed the experimental and our deployment setting in order to have significant data in different scenarios. In the fourth section, we will show the differences between the two frameworks under comparison in terms of experimental results analyzing the performance gap. Finally, in the Conclusions we indicate possible future works on LBS direction in terms of scalability, security also exploring new possible technologies for indoor location and possible evolutions in industry scenario.

II. Background

This section sketches related work on the Beacon protocol and the most relevant connectivity protocol implementations using the beacon protocol. In particular, we will mainly focus on works exploiting the Beacon protocol in conjunction with Bluetooth Low Energy.

A. Beacon Protocol

The beacon protocol was released for the first time by Apple and soon became a representative new system for location advertising and indoor location. In particular, Apple presents iBeacon which is a smart indoor positioning system delivered by Apple with iOS 7 which uses Bluetooth low energy for proximity advertisement of beacon presence. Apple released an SDK integrated with the iOS location framework for iBeacon detection and interaction. Apple has also released the beacon protocol specification in order to leave it open to every developer willing to implement it and to work with proximity sensor and to use beacon protocol for indoor location. iBeacon was very appreciated but, unfortunately, can only be supported by Apple devices. Radius network after the reaction of the community and the discovery of new industrial scenarios proposed its implementation of the beacon protocol called AltBeacon. In 2014 Google launched a new implementation of the beacon protocol aiming to expand the utilization of beacons for URL advertising using its proximity advertising capabilities. A year after Google proposed Eddystone, an extension of the beacon protocol which does not change the protocol of data transmission itself but only the data packet that could be sent (or received) adding a new category.

The beacon protocol is a simple data transfer protocol, where every implementation adds some specific fields in the data packet. The common fields are UUID, Major, and Minor. UUID is a string which could represent the organization who owns the devices or vendor itself. UUID, for instance, could be used for signaling when a device is entering to a location such as a museum. Major could be programmable and represent a set of beacons, could be used to identify a group of beacons in a specific sector of a museum. Minor is also programmable and could identify a subset (or a single) beacon of a Major group in order to advertise the device when is entering to a specific area, for instance, when is in front of a statue in a museum. Every implementation of the beacon protocol defines a threshold of RSSI that allows determining whether the device can be considered as "entered" or "not entered" within a target area.

The beacon protocol does not specify any underlying connectivity technology, allowing many implementations on top of different technologies. The main technologies used in hardware implementations for beacon protocol are Bluetooth Low Energy, Wi-Fi Aware, Ultrasound, and also a mix of previous technologies.

B. Beacon BLE

Bluetooth Low Energy (BLE or formerly Bluetooth Smart) was designed by Bluetooth Special Interest Group (Bluetooth SIG) for application scenarios like fitness, health, beacon and indoor location, security and most of all appliances with mid-low range communication without the necessity of high bandwidth. BLE could coexist with "Classic" Bluetooth implementation and aims to get same network capabilities with less power consumption.

BLE target market includes smart home, health care where low power consumption and very long autonomy has to be achieved feeding devices with the smallest battery commercially available; also, devices with BLE are very small and very cost-effective in the target use case. One of the main applications of BLE is health care with wireless measurement systems and proximity advertisement. BLE could be managed using devices Software Development Kit (SDK) requesting special features if these are supported. Bluetooth Smart could be also managed using a specific framework which internally requests and uses it. Beacon BLE are hardware transmitters, a class of BLE devices, that broadcast their identifier through

Bluetooth Smart in order to be compliant with beacon protocol. Beacon BLE are mainly used for location-aware advertising and indoor tracking.

Beacons have a lot of usages in normal life, for example, Sruthi Menon et al. [5] proposed an iBeacon-based system for Smart work with task automatic task dispatching. Every employee position has an iBeacon identified with UUID (Major and Minor) and the association with the employee is registered on the remote server. When the employee approaches its desk, the application on its device alter the presence state on the remote server in order to notify the arrival to manager; furthermore, the server pushes to the employee's device the tasks that it must do. The system also provides an internal chat between employees and manager.

Beacons are also the main component of smart home services, Louay Bassbouss et al. [6] have proposed an intelligent and reactive system for interaction with Smart TV and User's smartphone. The system places an iBeacon near the Smart TV in order to advertise the user device when it reaches the beacon's range, the device receives the advertise and reacts opening the Smart TV companion application. Furthermore, the system takes into account the possibility of handling multiple devices on the Smart TV side and handling custom advertisement if the user "knows" other TV like the one that is near to him.

BLE beacons fill the gap between traditional tracking systems like Wi-Fi or GPS for indoor location, also addresses other challenge related to a human in a unique way. For instance Yukiharu Motohashi et al. [7] uses iBeacon (and device sensor) for indoor tracking and pattern recognition in order to determine users next move in terms of internal movement for cooperation with advertising systems. For instance, if a user stands still in front of an opera the system will send to the device the opera description. Another example, which is currently in production state, is Casa Batllo [8]. In this system, there a hundred of beacons for indoor tracking, which guarantees an error of 50 centimeters; the LBS uses also augmented reality to show through device screen how was the original aspect of the house. In order to achieve this, the system relies on continuous device communication with the central server for an updated position and download related data.

Beacon BLE is also used in learning technologies, like the used proposed by Yukiharu Motohashi et al. [7] for museums. Yu-Lin Jeng et al. [9] have implemented a software for Situated Learning Application based on micro-location system offered by iBeacons. The beacons are located in precise environments when a user passes through the advertisement area if it has installed the relative application, the mobile device will show a badge which signals the possibility of learning a new part of the target language. When the application gets a new advertising from a beacon opens a virtual map with all definitions for the surrounding area. The application offers a different kind of teaching, from a single word to entire sentence based on user familiarity with the language.

iBeacons are also used in an industrial scenario like Warehouse Management System, Zhiheng Zhao et al. [10] uses beacon protocol (using iBeacon implementation) for forklift movements between bays in order to drive each forklift to bays and picking zones. As shown by Zhiheng Zhao et al. [10], iBeacons could be used in pairing with other protocols like Wi-Fi. Furthermore, they use iBeacon, not as passive static indoor location point but as mobile advertise broadcaster and there are base stations with BLE and Wi-Fi in order to communicate with the remote ERP. Beacons are located on a picking unit and associated with a forklift when they get picked, while the material has been moved the beacons continuously broadcast their data in order to inform the ERP about their location.

Mission critical systems also exploit their needs for high accuracy with low power devices using beacon protocol. Mirza Muhammad Bilal Baig et al. [11] uses iBeacon for healthcare management, they rely on beacon protocol as indoor location system in order to track movements of an elder patient. The patient wears healthcare monitoring sensors in order to have health parameters monitored, data produced will be sent using BLE to the target device (the patient phone), when an anomaly is detected the device will retrieve location data checking which is the nearest beacon and send this plus the healthcare sensor data to the nearest doctor. The nearest doctor is chosen using privacy-preserving location algorithm, also the data are sent to an E-Health server in order to preserve the privacy of the patient. In order to detect anomalies, the system uses a modified basic Bayesian classifier on server part.

III. Experiment Design and Deployment setting

This section shows our experimental deployment, in terms of conducted tests and parameters considered, for comparing Apple Core Location framework and Estimote Monitoring. We have compared the two frameworks by considering two different use cases: *monitoring* and *ranging.*

The *monitoring* use case enables proximity advertising when a device enters into a beacon range, which depends on the beacon connectivity. In the monitoring use case, the concept of the *region* is defined which consists in an area where a device could be reached by the beacon for advertising. For instance, when a user with his devices approaches its desk where a beacon is located and his device enters into the beacon region the user's device will be advertised and could "react" with a specific application. A Region could also be the intersection of one or more beacons advertisements and the application on the device can intercept events, like the entrance into the region or the exit from the region and can launch specific actions according to the specific occurred event. This enables an event-driven programming model also called proximity advertisement. In the *ranging* use-case it is possible to determine a reliable relative device position or to have fine-grained data about which beacons are in the device's proximity and beacon's distance from the device. For establishing its position a device tracks every beacon in its range. For instance, with *ranging* a location-based service, like an indoor navigation system, could be designed and implemented. In particular, ranging relies on the *region* concept and assumes a constant interaction with beacons and their regions, in order to have more information and detailed device movements. In order to properly exploit the *ranging* mechanism, a system must have a basic knowledge of the overall environment.

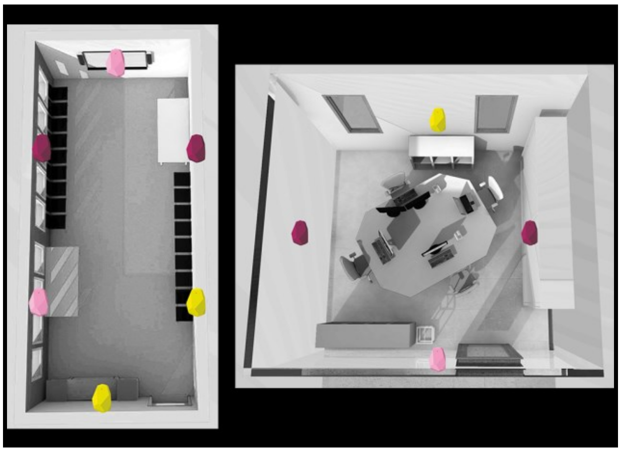

Fig. 1 Deployment of beacons: on the left, the deployment in the free noise room, on the right the deployment in the noisy room.

The realized deployment environment consists of two different rooms with different dimensions and internal organization and different electromagnetic interference. The BLE is a radio frequency wave, which operates in 2.4 GHz Industrial Scientific Medical (ISM) band [12]; that band is also used by other systems like 802.11 b/g which could easily interfere. Also, a powerful source of interference could cause a high error rate in measurements. The first room is rectangular and is 55 square meters big and there is no source of interference. The second room, instead, is smaller than the first one and has some electronic devices like laptops and mobile phones that could interfere using their Wi-Fi antenna.

In the first room, the one without any interference, we deployed six beacons, four on the long side (two on each side) and two on the short side (one each). In the second room, the smaller one, only four beacons are deployed because the room was smaller and the beacons transmission range cover the whole room (Fig. 1).

We have developed some applications in order to test respectively *monitoring* and *ranging* with Core Location Framework and Estimote Monitoring. The applications developed for *monitoring* tests will only log data, such as entering and exiting from a region and a timestamp in order to detect when the action is triggered. The applications for *ranging,* instead, were implemented in order to be compliant with the indoor location paradigm; we have defined the room as a new location within the application code and then the application interact with it using framework-dependent features. The applications complexity is higher than the *monitoring* one because of requirements of indoor location such as the need to have environment knowledge and region tracking.

The reliability and performance of each framework, regarding monitoring, were tested measuring the delay of the device between the device entrance and exit from a region; the measurements were taken logging when the correspondent entrance and exit event are thrown. Ranging, instead, was tested using indoor location paradigm. The applications "knows" the entire area and will log the relative position given by the framework; how the framework retrieves the position is implementation dependent.

A. *Core Location Framework iOS*

The Core Location Framework in iOS provides API and services in order to determine a reliable location wherever the device is, indoor or outdoor using device's sensors. It exposes API in terms of classes for interact with Outdoor Location, Compass, Beacon hiding all primitives for sensor use and tuning. The exposed classes follow the single responsibility principle, so there are API for iBeacon interaction, Geocoding and so on and so forth.

The classes for interaction with iBeacon are *CLBeacon* and *CLBeaconRegion* which implements and exposes methods for interaction with them and read the advertised data. Core Location Framework enables also, the capability of turn an iOS device into an iBeacon itself generating a UUID and "deploying" a beacon region sending the custom advertisement.

To implement monitoring with the iOS Framework, we use the *CLLocationManager* [13] class. This class allows to monitor one or more regions (identified by a combination of UUID, major and minor values of beacons) and raises events like entering in a region, the events will be caught according to delegation pattern. It is important to remember that with this method the region limit is identified by the strength of the signals of the beacons. Starting monitoring a region can be achieved with *startMonitoring(for: region)* method of the CLLocationManager class.

Also, the ranging technique can be implemented with the *CLLocationManager* class, using the *startRangingBeacons(in: region)* method and some other specific methods and supporting class (for example, to develop an indoor location system *CLFloor* class could be useful).

In order to have a proper detection of entering and exiting from a region, the Core Location Framework allows the definition of proper callbacks using iOS delegation semantics. In fact, entering and exiting region, events are delivered to the locationManager(_:didEnterRegion:) and locationManager(_:didExitRegion:) methods of class delegate, while in ranging technique the CLLocationManager object reports any encountered beacons to its delegate by calling the locationManager(_:didRangeBeacons:in:) method.

B. *Estimote Monitoring*

The Estimote Monitoring framework is a beacon protocol-centric framework, developed for interaction with beacons. This framework is available for iOS and Android and exposes uniform API for interaction with beacons. Estimote Monitoring could work with all implementations of beacon protocol and all devices which implements (or support) one target protocol.

The framework allows different types of interaction with beacons including a background monitoring service in order to discover new beacons while moving and collect precise RSSI for indoor location purposes.

The Estimote framework has a manager-based implementation, one manager for beacon-interaction technique. Each of these managers exposes appropriate methods for beacon handling depending on the chosen context.

TABLE I: QUALITATIVE COMPARISON

	Core Location iOS	*Estimote Framework*
Range of trigger	Given by RSSI power of each beacon	Software designed (anyway RSSI value must be strong enough)
Maximum number of regions monitored at the same time	Not more than 20	Unlimited regions
Delay in exit events	Up to 30 seconds, due to prevent "false" positives	Exit events without a fixed time threshold

Estimote Framework exposes *ESTMonitoringV2Manager* [14], which enables the monitoring of deployed beacons also by their single identifier in addition to the region system. The *startMonitoring(identifiers:)* method start the scanning and is possible to choose the trigger distance of the event. These events are delivered to the monitoringManager (didEnterDesiredRangeOfBeaconWithIdentifier:) and monitoringManager(didExitDesiredRangeOfBeaconWithIdentifier:) methods of ESTMonitoringV2Manager delegate.

The ranging technique is implemented by the class *EILIndoorLocationManager* [15] and the delegate method *indoorLocationManager(didUpdatePosition position:)* will detect the change of position of the application inside the room, allowing developers trigger specific actions based on user indoor position.

IV. EXPERIMENTAL RESULTS

The conducted experiment consists of about two hundred tests spread in different days for both use cases. We made first *monitoring* testing in the two rooms indicated above, in order to have a complete scenario.

A first implementation result has evidenced how Estimote Monitoring handles better RSSI. Apple's native solution uses RSSI values of the beacons to determine if a user enters or exits a region; due to interferences generated by the environment, the value can be imprecise, which means that events could be triggered at the wrong time or not triggered at all. In order to avoid that, iOS framework uses a delay of 30 seconds from the effective device exit of the region to trigger the event, avoiding false positives in RSSI values but introducing some significant delay. Estimote Monitoring uses proprietary algorithms to enhance performances, allowing to trigger events at chosen distances and to detect events without any delay avoiding also false positive detection (see Table I).

In addition, Table I shows a qualitative comparison between the two frameworks, taking into account parameters for framework scalability, performance, and flexibility. The range of trigger parameters is mainly focused on when each framework will trigger an event, based on distance (and relative RSSI) from the beacon. The maximum number of regions monitored at the same time represents how many regions could be under the monitoring of the two frameworks.

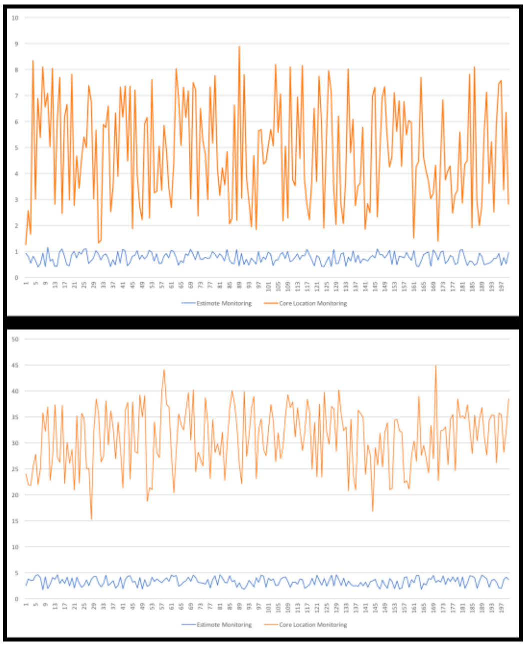

Fig 2. Results of experiments of entering (up) and exiting (down) a region in a free noise room. In orange with Core Location Framework, in blue with Estimote Monitoring.

Delay in exit events parameters shows how the frameworks handle the exit event from a region.

A. Monitoring

Apple's Core Location Framework results are not so encouraging for *monitoring* phase. Entering in a region in the room without noise with Apple's solution shows between two and twenty seconds, while for exiting a region are between twenty and forty seconds. Our tests with Core Location Framework also show 0the concrete possibility of false-positive event triggering. Estimote framework guarantees, instead, much better results than Core Location Framework. Entering into a region takes less than one second in the noise-free room and about ten seconds in the noisy room. The decision of Estimote's developers to trigger the exit event without any delay drove the experiment results below five and twenty seconds, respectively for the free-noise room and for the noisy room. In Fig. 2 you can see the difference between entering and exiting a region in the free noise room.

As you can see from Fig. 2, Estimote Monitoring enter and exit reaction time is better than Core Location Framework, leading us to avoid any other test with Apple's framework.

B. Ranging

We decided not to test *ranging* with Apple Core Location Framework being results for monitoring discouraging as shown in Section IV.A. On the contrary, we tested Estimote Framework for ranging. The experiment shows that Estimote Framework results for indoor location are accurate for both rooms. The highest error is about half a meter in both use cases, this represents an interesting result, especially in comparison with other techniques such as Wi-Fi indoor localization. Fig. 3 shows indoor location experimental results in ranging use case, which represent the most feasible

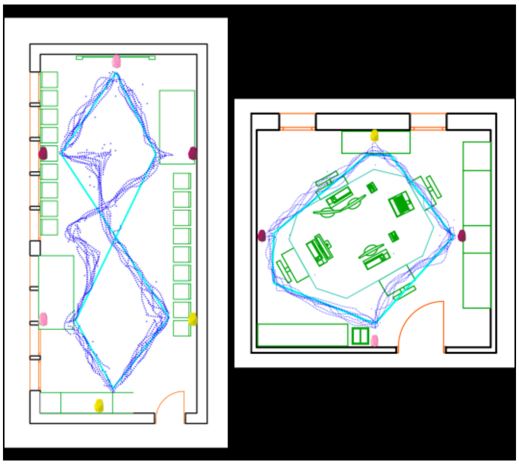

Fig 3. Indoor tracking with Estimote Indoor Location Framework. In light blue the effective path around the room, in blue the position estimated by the application. On the left in a free noise room and on the right in a noisy room.

representation of user movement inside a room with points of interest. Our testbed for a noisy environment was smaller than common use cases, like entire an entire wing of a museum, but shows how reliable and effective is the Estimote framework. We chose the path in the noisy room in order to have the most realistic simulation, regarding noise-free room we chose the most difficult path in order to test multiple enter-exit patterns from different regions.

V. CONCLUSION AND FUTURE WORK

We have deeply compared the Estimote Monitoring and Apple Core Location frameworks in order to have a reliable performance evaluation in all the possible use cases and we can conclude that the first framework demonstrate to be more mature.

In fact, experimental results show that Estimote framework is more reliable in the 80% of cases, in the remaining 20% is equal to Apple Core Location framework. The experiments have also evidenced the high reliability, and production readiness for Estimote monitoring in noisy environments regarding event triggering and monitoring use cases. Estimote framework is also more reliable in an indoor location use case, offering also a simpler development mechanism through its manager-based architecture; Core Location framework, instead, is a general purpose location framework offering poor performance and is not so simple to develop indoor location applications.

Nevertheless, in both use cases, exploiting the capability of each sensor on a modern device could bring a significant improvement in the application efficiency and accuracy; this could also lead to a better user experience.

Our work could be expanded in many different directions from different points of view, from a research point of view and industrial point of view. Regarding the research perspective, we want to investigate the security aspects by including also privacy and higher precision. For instance, the system could add more beacons and have higher precision making it more scalable and accurate, and privacy controls can be introduced enabling data sharing policies at the device level. From an industrial point of view, we want to add more exciting features such as interoperability with a smart home system, enabling, for instance, the possibility for the system to turn on the light when the user enters in a room.

REFERENCES

[1] Active Bat: http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/

[2] Active Badge: http://www.cl.cam.ac.uk/research/dtg/attarchive/ab.html

[3] Paolo Bellavista, Axel Kupper, and Sumi Helal, "Location-Based Services: Back to the Future", *Pervasive Computing magazine*, 2012

[4] Md Sayedul Aman, Haowen Jiang, Cuyler Quint, Kumar Yelamarthi, Ahmed Abdelgawad "Reliability Evaluation of iBeacon for MicroLocalization", *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),* 2016

[5] Sruthi Menon, Anu George, Nehal Mathew, Vishnu Vivek and Mrs.Jisha John, "SMART WORKPLACE - Using iBeacon", *International Conference on Networks & Advances in Computational Technologies,* 2017.

[6] Louay Bassbouss, Görkem Güçlü, Stephan Steglich, "Towards a remote launch mechanism of TV companion applications using iBeacon", *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*

[7] Yukiharu Motohashi and Eri Sato-Shimokawara, Rosanna Yuen-Yan Chan and Nan Zhou, Toru Yamaguchi, "Using iBeacon to Detect User Behavior from Indoor Physical Movement", *IEEE/SICE International Symposium,* 2017.

[8] Casa Batllo AR, http://www.idgconnect.com/abstract/10419/seeing-gaudi-casa-batll-augmented-reality

[9] Yu-Lin Jeng, Mu-Lan Hong, Hsin-Tse Lu, "The Development of the Mobile Situated Learning Application Based on Micro-location Technology", *6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)* 2017

[10] Zhiheng Zhao, Ji Fang, George Q Huang, Mengdi Zhang, "iBeacon Enabled Indoor Positioning for Warehouse Management" *4th International Symposium on Computational and Business Intelligence,* 2016.

[11] Mirza Muhammad Bilal Baig, Muhammad Taha Jilani, "An iBeacon based Real-time Context-Aware", 2017.

[12] Bluetooth Core Version 5.0, https://www.bluetooth.com/specifications/bluetooth-core-specification

[13] Estimote location manager library: https://developer.apple.com/documentation/corelocation/cllocationmanager

[14] Estimote monitoring library: https://estimote.github.io/iOS-SDK/Classes/ESTMonitoringV2Manager.html

[15] Estimote indoor location library: https://estimote.github.io/iOS-Indoor-SDK/Classes/EILIndoorLocationManager.html