

Bluetooth Low Energy Supported Indoor Location

Ricardo Martins ricardo.pachecomartins@gmail.com Instituto Superior Técnico

Abstract—Abstract

I. INTRODUCTION

Outdoor positioning systems have greatly evolved in the past with the global positioning system (GPS) as its most important system. Since GPS is an outdoor position systems and is based on a network of satellite, when the required scenario for position tracking is inside a building, new constraints are presented onto the process such as the attenuation and reflection of eletromagnetic waves upon collision with building walls and obstacles [1]. As such there was a need to find reliable indoor systems that by nature would reduce the impact of some of the mentioned constraints.

In order to understand indoor position there is a need to understand the full scope of variables that comes to surface when moving from outdoor to indoor. When developing an indoor system there is a need to make sure that it can tackle the challenges such as small space dimension which reforce the need for higher precision, a higher probability of non existent line of sight, influence of obstacles such as walls, furniture, moveable objects such as doors and human beings[2], [3]. All of the previously mentioned affect the way electromagnetic waves propagate in an indoor environment leading to problems related to severe multipath and reflection on existent surfaces [4]. Besides propagation challenges, there are energy consumption, accuracy and deployment costs that play a critical role in deciding the viability of a proposed indoor location technique. Another interesting challenges that also surfaces is the means to describe a location. When dealing with outdoor location systems, the provided location is always characterized by two values, latitude and longitude, since the existant environment, i.e. the planet Earth, is always the same and as such the coordinates system can be relative to it. In indoor systems, the surrounding environment can be of many shapes and as such different ways of representing the data are necessary. If one thinks about providing indoor location on an office, the precise location isn't as relevant as just knowing the general location, i.e. a building specific description of the location in the form of building, floor and room. On other environments such as supermarkets or even in the previous example, where the objective is to provide more a precise location description, a cartesian coordinate system (x,y) is required.

With the evolution of mobile devices there has been a sizeable number of different technologies that can possibly be used in indoor locationing [5], [6] such as GPS-based technologies, using high sensitivity antenas to overcome GPS's indoor issues, RFID , Wireless LAN and Bluetooth among

others, allowing even for hibrid systems which make use of more than one of the technologies mentioned above.

We are now in an era dominated by smartphones and as such they have became the central piece of several indoor location system. Smartphone's evolution allowed for developing system which rely on its sensors and processing capacities to present a solution which isn't dependant on specific hardware. With such a dependancy, the first condition that is imposed onto the systems is the compatibility with smartphones, i.e. the required sensors needs to exist on the generic hardware of smartphones. Once this barrier is surpassed, these generic system's are immediately faced with three fundamental questions which will define the architecture of the system and which can be seen on figure 1. The first question, Technology, defines the technology which is to be used in conjunction with the smartphone and consequently the way that location data is to be collected. There is a wide range of possibilities for this choice, be it BLE beacons, Wi-Fi Access points, LED lamps or just the microphone for sound collection, and each has an impact on the way that the system functions and on its performance. The second question is the location algorithm, which fundamentally depends on the target requirements of the system, if it's required to provide accurate location of a user or if a more descriptive location, such as the room in which the user is located, is enough. The third and last question is about which way will the location be described, with the existant possibilities having previously been presented.

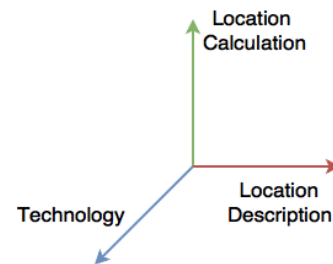


Fig. 1: Three fundamental choices in Indoor systems

When taking into consideration every possible technology and existant algorithm that are capable of being applied on indoor location, we find ourselves with a wide variaty of solutions. This high number of variations has led to an increase on the number of existing surveys which attempt to gather make a collection of what has been achieved in the field. Looking at this opportunity one saw the possibility of attempting to create a generic platform capable of being utilized by any existing indoor location system. In order to implement and test the solution there was a need to decide on which technology to

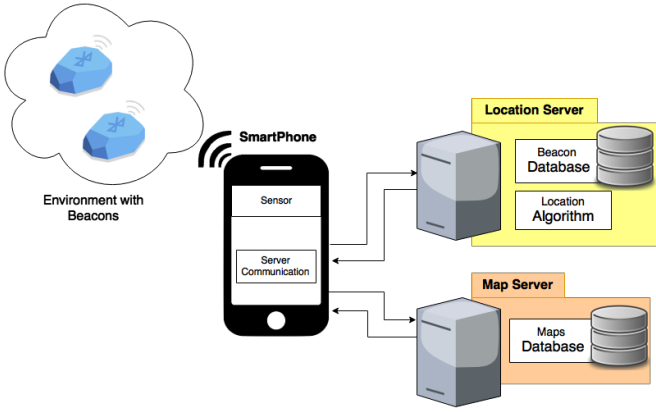


Fig. 2: Generic System's Architecture

be used and as such one opted by the bluetooth low energy.

The Idealized generic architecture, which can be seen on figure 2, was created with the intention of providing a common ground for any systems which planned to have an indoor location based on a smartphone, which allows interoperability among those who make use of it, while presenting a way to structure the system so that the impact on the smartphone is reduced. If one analyses it according to the three points presented in figure 1, one can say that by having dedicated serves for the location computation and the location description instead of having it locally on the smartphone, interoperability is highly improved. This setup allows for systems to implement whichever algorithm they intent for as long as the server is capable of distinguishing which algorithm is to be used, responsibility which is passed onto the smartphone. In scenarios of interoperability, the smartphone should be able to use any of its sensors to capture data, dealing with the technology component, and tag them before sending it for computation on the server. The location description is contained on the map server and it is up to the system to choose which one to utilise, be it self-created or owned by a third party such as google maps or openstreet maps. For as long as the smartphone is aware of the type of location that he is to receive, interoperability should be achieved.

This paper is structured in a way that in II an overview of existent indoor location systems, generic and dedicated, is given, followed by an analyses on the existent location techniques and the ways to decribe a location. Section ?? overviews the architecture of the proposed generic system, while section ?? presents the finalized system by analyzing the utilized technologies. Section ?? analyzes the different aspects of the presented solution in terms of energetic efficiency, accuracy and response timings, finalizing in section IV by overiewing what could be carried out in order to further develop the existing work and by concluding the paper.

II. RELATED WORD

In this chapter one will first analyse the systems that layed the foundations of indoor location systems in section II-A, in order to study systems that were fully dependant on having specific hardware for obtaining user location. Section II-B will

present the technologies that are compatible with modern day smartphones, analyse its pros and cons and mention existant projects based on them. The following section, ?? will describe the existent ways of obtaining the data that is required for inferring a user's location as well as the techniques that make use of them. To conclude, on section ??, one will present the existent ways of describing locations through system that make use of them.

A. Dedicated Technologies

In 1992 the Active badge system [7] was presented as an infrared solution which was capable of provided room-based position tracking, due to infrared signals being unable to travel through walls. This system imposed users to carry with them an ID card equipped with an IR LED which emitted a signal each fifteen seconds [8], [9]. The long signal frequency was due to the chance of collision of signals from multiple users in the same location while also having in mind energy cost which need to be kept as low as possible. One of the downsides was the decrease in accuracy as the user's location is only known at best to a 15-second window. The user's location was obtained through the implementation of a network of sensors which act as receivers and then forward the obtained information to the master station which would take care of polling the sensors, processing data and then making it available to users. One of the biggest issues that were raised with this system was the privacy concerns raised by the clients due to its object tracking nature.

At the beggining of the twentieth century, two remarkable projects were presented, Bat system [10] and RADAR[11]. The first one, Bat Ultrasonic Location System, was a system capable of tracking various objects, each with a small wireless device called bat [12], [13]. A bat consisted of a radio transmitter with a unique identifier and was ment to be carried by personnel or attached to objects. For locating a bat, the base station periodically transmitted a signal with a single ID which caused the corresponding bat to emmit an ultrasound pulse. This pulse was then captured by an existing grid of receivers, which were placed on the ceiling of the areas to be covered, which recorded the time of arrival of the ultrasound. Since the speed of sound in the air is known, the system was capable of converting the time of flight of the ultrasound into distances, in order to obtain the bat's location.

The Radar system is a RF-based system to locate and track users inside buildings. The system is divided into two phases, an offline phase used for data collection and a real-time phase where a location is obtained [14]. For data collection the mobile application and the base stations are synchronized, in order to be able to obtain time stamps and then the mobile user periodically emits UDP packets to the base stations (BS). Each BS records the RF signal strength measurement together with the time stamp, in order to create a fingerprint of the map. The real-time phase utilises the real-time user's packet associated RF signal strength, which is obtained at each bs and later on forwarded to the central computer where the computation is made, and examines it in order to find the best fit for the current transmitter position.

The last remarkable project, Cricket, managed to tackle some of the problems presented in the last mentioned systems [15]. Each node in the cricket system is composed of a RF transceiver and hardware capable of generating and receiving ultrasonic signals. There are two types of nodes, beacons, which are fixed reference points and are attached to the ceiling or walls of the building, and listeners which are attached to the objects that need to be tracked. Each beacon periodically transmits a RF signal message containing beacon specific information, such as unique ID and beacon position. Whenever a RF signal is transmitted, an ultrasonic pulse is also emitted which enables listeners to measure their distance to the beacons by using the time difference of arrival times of the RF and ultrasonic signals. Each listener utilizes the RF signal's beacon information alongside the obtain distances to beacons to compute their space position and orientation. Crickets Architecture allowed to solve the user privacy, decentralized and ease of deployment issues there were present in the remaining projects [16].

B. Generic Technologies

Wi-Fi is a technology that can be used to estimate the location of a mobile user that resides inside the network. Nowadays Wi-Fi positioning systems have become the most widespread approach for indoor location systems since Wi-Fi access points are readily available in many indoor environments and since all smartphones are equipped with a Wi-Fi antenna they can be located without the need of installing extra software or manipulating the hardware. One issue of Wi-Fi signals is that they suffer attenuation from static environment such as walls and movement of furniture and doors. The most widely location technique used is the radio signal strength indicator (RSSI), which suffers from severe multi-path effects leading to propagation model failures and as such inaccuracy in distance measurement. With these problems in mind a technique called RSSI-based fingerprinting is often used in order to improve performance, the same technique used in the Radar system.

Infrared (IR) systems are one of the most common position system that utilise wireless technology that has been used to track objects or people. IR wavelengths are invisible to the human eye under most circumstances, making this technology less intrusive than those which are visible. This technology is completely available in all smartphones since it makes use of the camera for data input, requiring only line-of-sight communication between receiver and transmitter, preferably without interference from strong light sources. One example is Epsilon [17], which makes use of Light-emitting Diode (LED) and light sensors. Epsilon presents a solution which is easy to use in today's world due to the availability of LED lighting in indoor environments, with the only requirement being the adaption of the existing infrastructure to the required by the system. In order for the system to work, it required the mobile user to carry a device with an incorporated light sensor, which is the case of pretty much any existent smartphone.

Bluetooth is a wireless technology that was created in 1994 with the objective of replacing cables connecting fixed or portable devices. The Bluetooth Low Energy protocol was

introduced with the Bluetooth Core Specification version 4 (also called Bluetooth Smart) [18] and it stood out for its lower power consumption, lower complexity and lower cost, while allowing for device discovery, connection establishment and connection mechanisms.

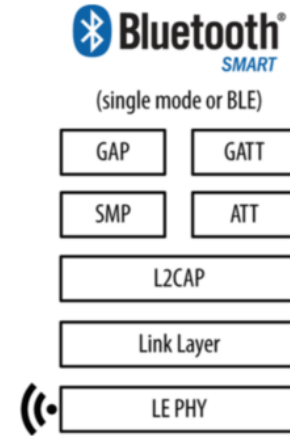


Fig. 3: Bluetooth low energy architecture

When looking at what's possible to achieve using the BLE technology there is the example of Apple's creation iBeacon [19] which was presented in 2013 with the purpose of implementing proximity sensing systems. The device is capable of playing on the broadcaster role and as such its objective is to send nearby compatible receivers certain information. Some examples of application are to track customers or trigger location-based actions on devices such as push notifications or checking in on social media, with practical cases such as the usage of iBeacons by McDonalds to offer special offers to their customers in their fast-food stores. An indoor location system utilizing this technology was presented by Jingjing Yang et al [20], where these devices were used to indicate a patient of his whereabouts through the proximity sensing proprieties and this information was later transferred over to a server in order to give clients a variety of different services, from patient counting, to nearby department's information and offer indoor guidance to the nearest available bed.

When utilizing BLE for indoor location the usual metric used to calculate distances is the RSSI. This metric within the context of bluetooth brings to surface several issues such as the fact that RSSI as a metric is very accurate only when the target is within a meter of the beacon, since the value decreases as the inverse of the square of the distance to the beacon. As such when developing solutions for indoor location that require system with high accuracy capable of tracking moving objects, the usage of RSSI can't be utilized without further work. Faragher et al [21] tackled one of the techniques used to improve BLE system's accuracy, fingerprinting, by verifying the effects caused by the device deployment density within the required location. This experiment also puts into evidence one of the downsides of the bluetooth technology being that its scalability is low, besides requiring higher density in order to increase accuracy, due to their low range any need to increase coverage leads to increased costs.

Zonith [22] introduced a bluetooth based location system with the objective of tracking the position of workers in dangerous environments. Any device registered in the zonith implemented network would be continuously tracked and accounted for in each of the system's functionalities such as, sounding an alarm whenever a lone worker doesn't move or responde within a time interval (Lone worker protection) or providing a quick an precise location of any worker that has requested for help. This system's installation requires planing of the best locations to place the beacons and number required of beacons in order to be able to provide enough coverage and make sure the system provides the required quality.

C. Location Calculation techniques

The location of a user can be obtained through several different techniques, each possibility leads to different levels of accuracy and hardware and computacional costs. The first technique would be proximity detection which has Cell of Origin (CoO) as its most famous technique and allows for room-based detection by assuming that the mobile target is at the cell with the strongest receiving power. This technique is widely used by system using RFID, Bluetooth and Infrared and has low deployment costs as the number of beacons required is small, at least one per room. The following technique would be triangulation which uses the geometric properties of the triangle to obtain a position. The metrics utilized to obtain the values used in the position's calculation can be angle-based, which have high implementation costs due to the required antenna's complexity; time-based, using Time-of-flight (ToF) or Round Trip Time (RTT) to obtain measures of distance from beacons; signal property based, using received signal strength indicator (RSSI), although it's only possible of using this metric with radio signals; dead reckoning, which calculates the actual position utilizing the last determined position and incrementing based on estimate of the device's speed.

D. Location Description

In order to better understand the existent ways of describing locations it's better to look at existent solutions and the methods that they utilise. The most common method of describing location is through latitude and longitude, much like the gps does, as the biggest example that makes use of it is Google maps. Google maps, through its indoor maps platforms [23] allows integration of indoor maps onto their google maps. Since google maps uses this type of location description, the indoor component of it follows the same routes. An indoor map on the platform is inserted onto its original geographical location and in the case of having multiple floors, it is possible to navigate through them.

Another method is to consider the indoor map as the reference and utilise a cartesion coordinate system, x and y , to represent a location. Meridian, an indoor location system developed by HP [24] which presents functionalities such as route making and push notifications through its beacons, makes use of its platform for insertion of the indoor maps. As such there isn't the outdoor component that google indoor

maps had, making it possible to use a (x,y) system relative to the building map.

Although Meridian already provides some extra level of detail in comparison with google indoor maps, there is another example that should be mentioned called OpenStreetMaps. Although it started much like google, providing global data, due to its openness, many indoor projects surfaced such as Open-LevelUP [25]. This project makes use of OpenStreetMaps' current indoor tagging scheme [26], which is intended to describe in the most complete and simple way a building. This makes available the number of floors, the type of elemets (room, wall, corridor, etc) and its connectors, like doors and escalators or elevators, allowing to understand clearly the map that is being analysed.

III. GEFILOC

A. Deployment environment

In order to understand how a system based on this generic architecture, figure 2 has to be strutured, it is of relevance to analyse the system component's description and requirements that would allow it to be deployed. If one starts by analysing the beacons component, one can describe them as the reference points of the system which are in charge of providing nearby users with information relative to their surroundings. These beacons are always ready and available to the users and upon being contacted, they should inform the smartphone which has initiated the communication of the address of the beacon's associated location server. Since the beacon only communicates with the smartphone, it isn't required to have any other communication capacity other than that specific to its technology. The last requirement is that the beacon needs to be uniquely identified and have the address of its associated server, while being capable of transmitting that same information to the smartphone.

The smartphone service is the component that is in charge of communicating with both the beacons and the location server. This service is a software component of the smartphone application which is available to being called when a location is requested. Once a location is requested by the application, the service should communicate with nearby beacons through one of its sensors. A service is associated with one and only one technology, as such if the application is to support more than one technlogy, multiple services should be implemented. This condition is relevant since the required sensor depends on the chosen technology and because it allows the service to quickly associate the data that it is collecting with the technology, information which must be delivered to the location server. Once the beacon data has been collected, the same data is forwarded to the beacon's associated server, together with a tag representing the technology of the beacons. Upon obtaining a location, the service should forward the data to the application, which is responsible of later communicating with the map server. In terms of requirements, the service needs to have Wi-Fi or at least 3G available for communicating with both the location and maps server, and access to the sensor required to communicate with the nearby beacon, for BLE it would be the bluetooth antenna or for QR it would be the camera.

The Location server should have a database of all the devices associated to it, each with its exact location of the map. This location is dependent on the method for location description chosen, i.e. latitude/longitude or x/y. The server should also be able to take input from any of the supported technologies, depends on the beacons in the database, in order to output the user's location. From this description one can define the requirements of the location server as having in storage the beacons that are associated it as well as their location, being capable of communicating with the smartphone application and being in accordance with the map server, achieved through providing location description that is the same as that present in the maps server.

The Map server is responsible for providing the application with the maps that it has requested. The only decision concerning this server is the type of location description that is to be used and consequently the only requirement is that it has to be in accordance with the information on the location server.

B. Architecture

The solution presented in this paper was made with the objective of creating a generic indoor location system capable of being implemented using any existant indoor system created. The generic system's architecture is presented in figure 2 and it's divided in 4 parts: beacon, location server, map server and smartphone application.

The environment with beacons represents any form of element responsible for providing fixed reference points which are fundamental in calculating a user's position. A beacon needs to be compatible with the usage of a smartphone, i.e. the device's sensors need to be able to capture the data. Using the previously presented technologies as examples, these "beacons" could be in fact BLE beacons, Wi-Fi access points, LED lamps or even something as simple as sound, with the data capturing being made through the smartphone's available antennas, camera or microphone. The beacon represents a position in the indoor environments and as such it needs to be uniquely identified, be it through its characteristics or associated information. In addition to these informations, each beacon needs to have an associated location server and provide information about it to the smartphone application.

The location server is an external component where the information relative to the "beacons" utilised is stored. As such when a location request arrives from the smartphone, the received data can be translated onto physical locations which will later be used to compute the user's location. The algorithm used to obtain such location is entirely dependant on the system's creators. Once the location is obtain, it's sent back to the smartphone in addition to the location of the server where it should retrieve the associated map.

The map server represents the architectural block responsible for providing the maps associated to the location of the user, location which was obtained through the location service on the application. The used map representation is up to the system creater for as long as it is in accordance with the remaining parts of the system. This means that the location

provided by the location server needs to be something that is representable on the maps provided and capable of being comprehended by the smartphone application.

The smartphone application is the central piece of this architecture. It is responsible of discovering and communicating with the beacons existent in its environment through its sensors. This communication process allows the smartphone to obtain information of its surrounding and to obtain the address of the location server that is associated to the connected beacon, in order to later forward this same information to it. Upon communicating with the location server, the application expects to get back information relative to its positions as well as the address of the map server that is to be contacted. Through that address, the application is capable of obtaining the map relative to its positions and display the result to the user.

The presented architecture was structured in a way that it is scalable and it allowed for interoperability. The idea behind such an architecture would be let different indoor building use their own version of the system but a user with the associated smartphone application would be able to transit between buildings and use the readily available beacons to always obtain its location. This would be achieved through a common architecture that allowed for self-contained implementations that could be access through a generic smartphone application. In order to analyse this assumption it is necessary to look back on the already analysed figure 1, which represents the main components of an indoor systems. This architecture is capable of externalizing from the smartphone component all the remaining. By making the smartphone the central piece of communication, the beacons are required to the ones to provide location data. By having a server dedicated to computing the location of an user, the system developers are allowed to use whichever algorithm they wish, since there is no dependency to the application. They are also the only ones responsible for having the beacon associated data, for giving those same beacons the location of its server and for computing a location that is in accordance with how they wish to store their maps on the maps server. The last remaining component is the location description which is completely passed onto the map server and whose only requirement is for it to be in accordance with its associated location server, in terms of location description.

C. Implementation

The implementation presented in this paper was created by utilizing the generic indoor location system presented in section III-B and applying it with bluetooth low energy. The system's architecture is presented in figure 4 and is divided three parts: the bluetooth low energy device, in section III-C2 a description of the used technologies and the changes made are present; the server, whose functionalities and stored information are described in section III-C4; and the smartphone application, whose process is described in section III-C3 alongside figures that show the functional prototype. For each of these parts an explanation will be given, containing a description of each of its components specific to the presented system alongside the requirements for each to work.

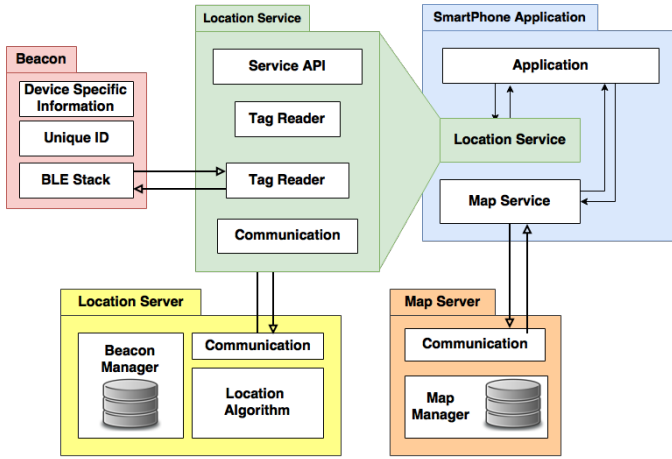


Fig. 4: System's Architecture

1) *Bluetooth Low energy*: When looking at the BLE's architecture, which can be seen at 3, the Generic Access Profile (GAP) is one of the most relevant since it's responsible for working in conjunction with Generic Attribute (GATT) to define the base functionality of BLE devices. The services that are made available by the GAP are: BLE device discovery, connection modes, security, authentication, association models and service discovery. In addition to this, it also defines four different roles to describe a device, allowing for the controllers to be optimized in function of the device's desired roles: Broadcaster, role optimized for transmitter-only applications; Observer, role optimized for receiver-only applications and it's complementary to the broadcaster role; Peripheral, role optimized for devices that only want to support a single connection, allowing for a much less complex controller due to the fact that it only needs to support the slave role and not the master one; Central, role supports multiple connections and functions as the initiator for all of them. These connections are all made with Peripheral devices and its controller must support the master role in a connection and allow for more complex functions, in comparison to the remaining roles.

Another important component is the Attribute Protocol (ATT) Protocol which is responsible for implementing the Peer-to-peer (P2P) protocol between an attribute server and client. This communication happens in a dedicated fixed channel and a server can send through it responses, notifications and indications, while the client can send requests, commands and confirmations. The ATT allows the clients to read and write values of attributes on a peer device acting as a server.

The last component is the Generic Attribute (GATT) Profile which is responsible for creating a framework for the ATT, in which it's represented the functionalities of an ATT server. This profile describes the hierarchy of services, characteristics and attributes existent in the server and provides an interface for discovering, reading, writing and indicating service characteristics and profiles.

Bluetooth low energy devices utilize profiles which define the required functionalities of the device. It also defines application behaviour and data formats and as such when two devices comply with all the requirements of a Bluetooth pro-

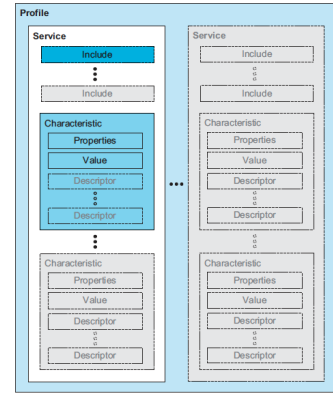


Fig. 5: Gatt-based profile hierarchy

file, application interoperability is achieved. Each Bluetooth profile describes its requirements necessary for devices to create a connection, to find available services and connection information required for making application level connections.

The base profile that any Bluetooth system needs to include is the GAP. From this point, any additional profile implemented will be a superset of GAP, where GATT is included and specifies the profile hierarchy, or the structure in which profile data is exchanged. Figure 5 shows the hierarchy in a Gatt-based profile, with the profile being the top level and services and characteristics below. A profile is composed by one or more services. A service is a collection of data and associated behaviors to accomplish a particular function or feature of a device or portions of a device. A service is composed of characteristics and/or references to other services. A Characteristic is a value that is used in a service that has properties and configuration information that describe how the value should be accessed as well as information on how to display the value. A characteristic is defined by its declaration, its properties, its value and may also be defined by its descriptor, which describes the value or permit configuration of the server relative to the value.

2) *BLE beacon*: The beacons that were utilized are Texas Instruments CC2650STK devices which can be visualized in figure 6. Alongside the device, which comes with a pre-installed bluetooth low energy program capable of giving information on each of its ten sensors through its predefined profiles, there is a texas smartphone application that can connect to a single device and read from its sensors. By using the texas Code Composer Studio (CSS), the pre-defined ble profile existent on the device could be altered. Upon further analysis of the profile, a characteristic was found for which the Universal Unique Identifier (UUID) of the service and the characteristic itself was found and as such this was the one that ended up being used to store the device's owner server's address. Since the device was already set to work as a peripheral and it now stored the information relevant to the system, there was no need to do further work.

3) *Smartphone service*: The Smartphone application was developed for Android using the Android Studio IDE. The Application is divided in two primary functional blocks, the Mock location Provider and the Google Maps Integrated



Fig. 6: TI cc2650stk sensortag

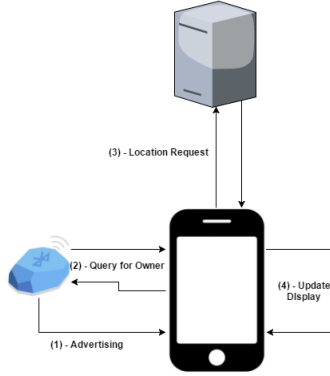


Fig. 7: Mock Location Provider Workflow

Display, as can be visualized in figure 4.

The Mock Location Provider is implemented as if it was a Location provider, such as gps. The application functions works as a listener to a Location provider, in this case it listens to the Mock Provider that was implemented. By implementing the whole process of obtaining a location inside a service (the mock provider), a new level of abstraction is added to the application. As such, whenever the application is signaled to obtain the user's current location, a request is made to the associated location provider and the application only need to listen for the answer that eventually arrives.

The Mock Location Provider incorporates the first three steps present in figure 7, which will now be explored individually. The first step indicates the gathering of information of the surroundings of the user's device. When a request is made to the provider, a scan for nearby bluetooth low energy devices is made which will put the smartphone in a state of listening for incoming ble advertisement packets. During this scanning period, each time a device is found, the advertisement is registered in a list, which have a duplication prevention mechanism implemented. Once the period is over, the provider has available a list of all the ble devices within range.

The second step involves taking the created list of devices, obtain a server address and forward the same list to it. Once the first step is completed, the provider will analyze each entry at a time. For each device the provider will attempt to respond to the caught advertisement packet, resulting in a created connection. Once the connection is created the provider asks for the available services of the paired device. Upon receiving an answer, the list of services is swoop while looking for the service with the wanted UUID. If the device doesn't have the UUID that the provider is looking for, it can assume that

the paired ble is not a beacon of our system, as such the connection is terminated. When the provider identifies that the device has the system's UUID, it requests the device to provide the service's existant characteristics. The provider will receive a list composed of the service's characteristics and it will search in it for the system's characteristics UUID, the one which contains the device's server's address. This search has the objective of confirming that the service existant in this device is indeed the one that was implemented for the system and not a device with another service that happened to have the same UUID. For any service outside those that are documented in the Bluetooth Special Interest Group (SIG), who have a specific UUID attached to them, the UUID is generated randomly and as such there is a small chance of collision. Once the wanted characteristic is found, the provider requests the device to read its value and stores the received value in a list. This list will contain the servers of the devices that were found, and for each address there will be a list corresponding to each device, and their corresponding rssi values, from the same owner. In order to quicken the previously described process, the provider keeps in cache the most recent contacted devices. Before attempting a connection, the provider confirms that the device isn't found in cache and when finishing a process, the associated device is inserted into the cache.

When every device has been contacted, a voting system is actioned which will decide from the list of servers which one it will send the collected information to. The voting system uses an exponential function in order to attribute a weight to each server.

The voting system was implemented with the objective providing a thin security layer by allowing multiple devices of the same server to overcome a single attacker's device which happened to be close to the user. After obtaining each server's values, the one with the highest value is chosen and sent the list with all the devices.

The Third step involves a simple client/server tcp interaction. The application starts off by formulating the message that it will later on send to the server, this message includes all pairs of device mac address and its associated rssi value captured by the application on the first step. Once the message is computed, the application attempts to create a connection with the server at the chosen address at the end of step two. With the connection established, the message is forwarded to the server and the application is put onto blocked state where it awaits for an answer. Upon arrival, the answer received is checked for valid location, its information is process and the connection is terminated. The information contained inside the received message, which was described in section III-C4, is then processed into the adequate class capable of storing a geographic location and the same is broadcasted from the mock location provider to its listener.

The Google Maps Integrated display is implemented using the Google Maps Android API. By using Google Maps it was possible to alleviate the weight on application since there wasn't need implement file transfer of indoor building's maps from each dedicated server to each request, which alleviated the servers aswell since there was no need to store its associated building's maps on it. Managing the maps was something

that was aswell fortunately unnecessary and as such all these features were provided by google maps service. By making this development choice, the system as whole became closer to the desired generic approach while making possible for seamless transition between indoor/outdoor maps. The only imposed restriction is related to the addition of new indoor maps onto the google maps, which is possible and well documented but dependent on a third party.

The Fourth step is called when the application receives a proper location from the request made onto the location provider. With the device's location known, a marker is placed on the map with the obtain coordinates (longitude, latitude), the camera is moved in order to be centralized on the position and fully displaying the indoor level map, and the menu visible on figure ?? is updated with the information that is bundled with the received location. In order to show the correct level on a multi level building, the "floor" information present in the menu is utilized. The API allows for obtaining a list of existant levels on which the maps' camera is focused and as such it's possible to find out to which level the provided locaiton belongs and make so that the application shows it.

4) *Server*: The webserver was implement in Python 3 programming language. The program implements a simple tcp server capable of receiving multiple request at the same time. Each request starts with information sent from an application which include a pair of MAC address and associated RSSI value for each ble device that the same application found. Afterwards the list of pairs is filtered in order to remove any existant devices that are not present in the server's database of devices.

Each server has a database that includes only ble devices. An entry (description of a device) in this database is composed by the device's mac address, its longitude and latitude and its building, floor and room name. In addition to the database, a server when initiated can store additional location info such as the server's street, number, zipcode, city and country, allowing this information to be transmitted to the client in order to offer an additional level of location description to the user. The whole location specific information can be visualised in figure 9.

Upon having filtered the initial list of pairs, the Cell of Origin (CoO) technique is applied by verifying which of the devices produced a stronger signal on the receptor. Upon obtaining the closest device an answer is sent to the application containing all of the information associated to the server and the selected device.

D. Evaluation

The pop-up menu was implemented to demonstrate the capacity of providing additional information associated with each location, be it geo-location taxonomy as it is currently implemented or possibly a description of the located room, an hyperlink of some sort or any other type of data that someone implemented this system would like to provide to its users.

The final state of the implemented system can be visualized in figure 8 and figure 9. The first displays the case of obtaining a location, where the marker has been placed and the camera zoomed

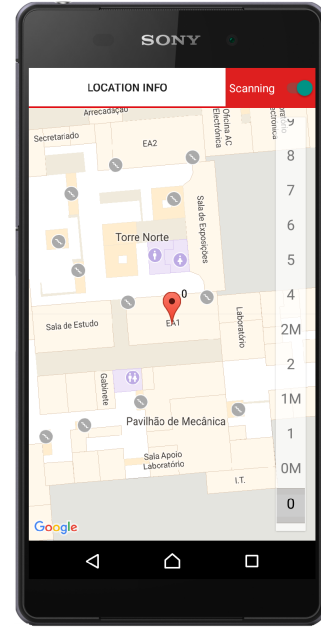


Fig. 8: Application screen showing a focused location on a room

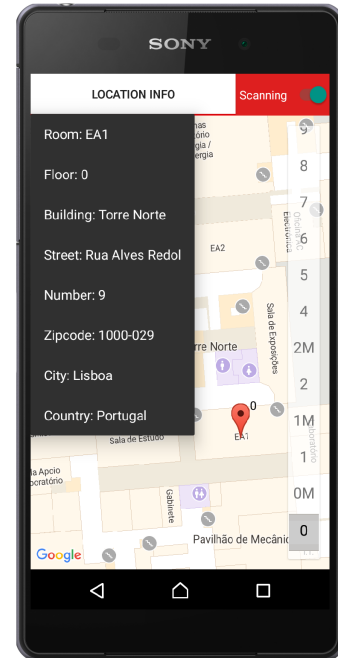


Fig. 9: Application screen showing additional information of location

- Energy Consumption in general
- Energy consumption vs Cycle time vs Accuracy?
- Time for request completion
- Data Requirements?

Although a complete implementation of the proposed architecture was achieved for BLE, one must indicate the required changes that would be necessary to make when change one or several parts of the system. The parts that can be changed are the three possibilities previously studied from figure 1. In case the technology is to be change, we will use QR code as an example, the affected areas are the beacons, the smartphone service and the location server. By changing the to QR code, the technology changes instantly as there is no more requirement for BLE beacons but instead there will be QR codes placed in fixed places. Each QR code would be required contain information about its associated server. The service would change the mechanism utilised to collect data, instead of listening for BLE beacons through its bluetooth antenna, it will open up the camera, so that the user can capture the QR code which would then be forwarded to the location server. The location server would have to be prepared for receiving the QR data and obtaining the location relative to that QR code.

If the system were to be changed on the location algorithm component, this would only require changes on the location server. The algorithms connection has only two connections points with the system, the data that it receives and the location it outputs. As such, for as long as the new algorithm makes use of the same data and in the end, once it has concluded, it outputs one location of the same format, no other issue should surface.

A modification on the location description utilised would have an impact on both the location and map server. With such a request, the map server is required to change completely the stored maps in order to support the new description method, while the changes on the location server are on the database, since the location of the beacons should change, and on the way the location algorithm processes these same locations.

IV. FUTURE WORK

- Conclusions

REFERENCES

- [1] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective."
- [2] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system."
- [3] K. A. Nuaimi and H. Kamel, "A survey of indoor positioning systems and algorithms," *Innovations in Information Technology (IIT)*, 2011.
- [4] H. Liu, "Survey of wireless indoor positioning techniques and systems."
- [5] R. Mautz, "Indoor positioning technologies," Master's thesis, ETH Zurich, February 2012.
- [6] H. Koyuncu and S. H. Yang, "A survey of indoor positioning and object locating systems."
- [7] [Online]. Available: <http://www.cl.cam.ac.uk/research/dtg/attarchive/ab.html>
- [8] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 59–68, August 1992.
- [9] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, October 1997.
- [10] [Online]. Available: <http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/>
- [11] [Online]. Available: <https://www.microsoft.com/en-us/research/project/radar>
- [12] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," 199.
- [13] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a sentient computing system," *IEEE Computer Magazine*, vol. 34, no. 8, pp. 50–56, August 2001.
- [14] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," March 2000.
- [15] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," 2009.
- [16] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, June 2005.
- [17] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system."
- [18] B. SIG, *Core Version 4.2*, 2014.
- [19] [Online]. Available: <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>
- [20] J. Yang, Z. Wang, and X. Zhang, "An ibeacon-based indoor positioning systems for hospitals," *International Journal of Smart Home*, vol. 9, no. 7, 2015.
- [21] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications."
- [22] [Online]. Available: http://media.teldio.com/pdfs/Teldio_Indoor%20Positioning%20White%20Paper_v2.1a.pdf
- [23] [Online]. Available: <https://www.google.com/maps/about/partners/indoormap/>
- [24] [Online]. Available: <http://meridianapps.com/sdks/>
- [25] [Online]. Available: <http://wiki.openstreetmap.org/wiki/OpenLevelUp>
- [26] [Online]. Available: http://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging