Internet of Things 13 (2021) 100334

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Discovering location based services: A unified approach for heterogeneous indoor localization systems

Francesco Furfari[a], Antonino Crivello[a], Paolo Baronti[a], Paolo Barsocchi[a], Michele Girolami[a], Filippo Palumbo[a,*], Darwin Quezada-Gaibor[b], Germán M. Mendoza Silva[b], Joaquín Torres-Sospedra[b,c]

[a] *Information Science and Technologies Institute - National Research Council (ISTI-CNR), Pisa, Italy*
[b] *Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain*
[c] *UBIK Geospatial Solutions S.L., Castellón, Spain*

ARTICLE INFO

Article history:
Received 20 May 2020
Revised 23 November 2020
Accepted 24 November 2020
Available online 1 December 2020

Keywords:
Indoor localization
Location based services
Context-awareness
Service discovery protocol

ABSTRACT

The technological solutions and communication capabilities offered by the Internet of Things paradigm, in terms of raising availability of wearable devices, the ubiquitous internet connection, and the presence on the market of service-oriented solutions, have allowed a wide proposal of Location Based Services (LBS). In a close future, we foresee that companies and service providers will have developed reliable solutions to address indoor positioning, as basis for useful location based services. These solutions will be different from each other and they will adopt different hardware and processing techniques. This paper describes the proposal of a unified approach for Indoor Localization Systems that enables the cooperation between heterogeneous solutions and their functional modules. To this end, we designed an integrated architecture that, abstracting its main components, allows a seamless interaction among them. Finally, we present a working prototype of such architecture, which is based on the popular Telegram application for Android, as an integration demonstrator. The integration of the three main phases –namely the discovery phase, the User Agent self-configuration, and the indoor map retrieval/rendering– demonstrates the feasibility of the proposed integrated architecture.

1. Introduction

In a world where devices can communicate among them thanks to the Internet of Things (IoT) paradigm, the presence of an architecture that allows the discovery and composition of systems built upon smart devices represents a key enabling factor for the real development and deployment of IoT applications and to ensure security, reliability, and efficiency [1].

The trend is towards increasingly smaller devices, a better energy efficiency, and a wider range of IoT-based services [2,3]. In this context, Location Based Services (LBSs) have emerged due to the ubiquitous Internet connectivity and the presence of cheap and easy-to-deploy Indoor Localization Systems (ILSs) [4]. Positioning represents an important field of application for

---

[*] Corresponding author.
E-mail address: filippo.palumbo@isti.cnr.it (F. Palumbo).

https://doi.org/10.1016/j.iot.2020.100334

IoT, for example in supply chain management and logistics (e.g. very early implementations with RFID chips among the first IoT applications ever developed [5,6]). But, besides being the ability of determining a position inside buildings one of the most important features of a service in that area, current developments clearly go far beyond a simple dot on a map [7,8]. In fact, several off-the-shelf solutions for indoor localization and positioning have been successfully proposed for complex [9] and crowded indoor environments –e.g. international airports [10,11], university campuses [12,13], hospitals [14,15], or museums [16,17]– where LBSs might be the key factor to bring together the user experience with market and/or public services. The localizing technologies appear to be mature enough to enable an ubiquitous deployment of LBSs as demonstrated by the raising interest of stakeholders, companies, and end-users on vertical solutions [18,19].

With this work, we propose the design and the prototype of an integrated solution for ILSs available in indoor environments. More specifically, we describe a software architecture designed to discover and to access ILSs by using open and standard technologies. This work is mainly motivated by two key observations:

- to limit the market fragmentation and the use of non-standardized technologies;
- to enhance the user experience and the seamless integration of different solutions.

Concerning the first point, we observe that currently, there is a reference guide for designing localization and proximity techniques and there is no indoor positioning technology widely-established for future deployments, i.e., there is not yet an overall satisfying solution for the IPS problem as stated by Brena et al. [20]. Every actor adopts its own solution in terms of positioning technologies and data processing and, more importantly, such systems are not designed to cooperate. As a result, we expect that the market for indoor systems will become quickly fragmented and characterized by incompatible solutions.

Secondly, we aim at increasing the overall user experience for location-based services with a solution designed to cooperate with multiple heterogeneous systems. We consider that such an approach offers several benefits with respect to closed solutions. e.g. increasing the accuracy of the system, increasing the reliability and the redundancy of the positioning, promoting the growth of a market of services based on indoor localization systems.

To this end, we describe an abstract architecture and we identify some key-components, namely the ILS, the Map Server and the User Agent. We describe how such key-components can cooperate and how a device can discover an ILS with a local and a global search. We finally describe our prototyping architecture based on the customization of the popular Telegram app. We adopted such an approach with the goal of offering to the end-user a well-known interaction paradigm, based on a chat system.

The solution we propose offers the following advantages:

- to discover the available location-based services in that area;
- to expose the provided capabilities and the required information by an ILS;
- to compose different ILSs in order to provide more accurate positioning information or LBSs with more capabilities;
- to manage the navigation service (e.g. maps rendering and path computation);
- to manage the *vertical handover* of the positioning service from one ILS to another;
- to manage the *horizontal handover* of the positioning service from one component to another in the same ILS based on the availability of devices and information in that area.

The remainder of the paper is organized as follows: we review the current literature and some reference commercial solutions in Section 2. Section 3 shows the integrated infrastructure to enable the discovery of maps, LBS, and ILS; Section 4 provides the details of the discovery mechanism; Section 5 describes the implementation details of the first prototype of the proposed integrated solution and, lastly, Section 6 draws some conclusions.

2. Related works

The IoT and machine-to-machine (M2M) communications provide a great occasion to many sectors. However, so far many researchers proposed vertically developed IoT/M2M solutions. These solutions typically use specifically customized software and hardware for a given sector, such as smart city, industry 4.0, etc. Since there already exist many IoT technologies in the market, global standards and inter-working solutions are critical to the success of the IoT [21–23]. In this context, our proposed architecture for ILS can be easily integrated as a service in any integrated platform able to manage multiple IoT devices [24,25]. Indeed, indoor positioning as part of ILSs is a key enabling technology for IoT applications [1], in particular in the field of LBSs. Different system have been proposed in the literature and on the market to address the need for this kind of information when building useful LBS scenarios such as guiding visitors/customers inside a shopping mall, convention centers, or museum, where conventional navigation technologies like GPS is not available [26–28]. Several different technologies and techniques have been used for indoor positioning, localization and navigation.[1] However, there is no ubiquitous ILSs –in contrast to GNSS outdoors– since most of the technologies and techniques are disjoint and they do not provide any interoperability mechanisms. This poses significant challenges about the needed of a proper integrated architecture for ILSs.

---

[1] Positioning and localization can be considered synonyms, positioning is usually referred to knowing one's own absolute coordinates while localization is more general and refers to knowing even the position of third entities (people or objects).

Kolodziej [29] already identified in 2004 that interoperability was a key pillar for developing LBS applications, whose communication, positioning, mapping, and software/services infrastructure needed to improve. Enhancing the modularity and standardization would enable LBS applications to reach the critical mass and become widely used, being stand-alone solutions unlikely to make a large impact on the marketplace.

Gratsias et al. [30] presented a taxonomy for LBS based on the mobility (e.g., stationary or mobile) of the query object (as user) and data object. They do not only identify the relevant applications when both objects are stationary (e.g., What-is-around, Routing and Find-the-nearest), but also Guide-me (Mobile & Stationary), Find-me (Stationary & Mobile) and Get-together (both Mobile). For each of the four application classes, they provided efficient algorithms targeting real-world applications.

Di Flora et al. [31] proposed an architecture for hybrid (indoor/outdoor) localization, that is independent to the location technology. Such architecture was designed not only to ensure compliance with upcoming standards and commercial devices/applications, but also to enable the interoperability with third-party services. In contrast to other open-approaches, they directly managed multiple location-sensors in an application-transparent way. Finally, despite identifying privacy/security as a core feature by implementing basic client-side functionality for selecting trusted infrastructures, they left this part to future work.

Kjærgaard [32] introduced an open taxonomy limited only to ILS based on fingerprinting. As several solutions had been already proposed, that taxonomy could help to understand the different available approaches by having the answers to relevant questions (e.g., deployment scale, estimation method used, temporal variation, data collection, among many others). It was built on top of a literature survey of 51 works that cover 30 different ILSs.

Wirola et al. [33] reviewed some relevant considerations and challenges of Indoor Positioning under three different stages: user case and service; architecture of the indoor positioning service; and technical realization. The authors concluded that the positioning architectural choice is not trivial but of great significance from the user experience point of view. Positioning error and consistency, as well as the time to first fix, are key features that must be considered when designing/developing an Indoor Localization System. Several service and architectural considerations were highlighted, with special focus on the position and navigation.

Ruiz-López et al. [34] performed an analytical survey on Indoor Positioning Systems, identifying that LBS could be designed as a Service-oriented Architectures (SOA). They argued that any positioning system should take benefit of the existing diversity of technologies and methods, so they should be designed to allow the integration of several positioning methods working, satisfying at the same time some non-functional (e.g., accuracy, complexity, responsiveness, robustness, scalability, among others) requirements.

Molina et al [11] proposed a multimodal indoor positioning system based on fingerprinting and covering wide areas in airports. In contrast to other traditional approaches, which rely on just one network technology, they integrated two technologies for accuracy improvement. They introduced a feature to allow merge or integrate additional technologies if available, which require modular programming and enhanced interoperability.

Lemic et al. [35] introduced a design and a prototypical implementation of a middleware architecture for fusing multiple sources of information. Through experiments in a real world setting, they achieve good results taking benefits, in terms of positioning accuracy, from the fusion of representative sources of location information deployed in the same indoor environment. A similar idea, presented in Stevenson et al. [36], describes a location model and an extensible framework for exploring location data's multifaceted representations and uses. Basically, authors' goal was to construct a framework that glued mapping space, positioning systems, and querying data, exposing a query layer. However, the real application of their inspiring proposal would force companies who provide location based systems to model their own systems sharing the same data representation. Graichen et al. introduced a map-framework where OpenStreetMap (OSM) data was not only used for rendering combined outdoor and indoor maps but also for calculating navigation routes and correcting indoor positioning estimates [37].

Zeinalipour-Yazti and Laoudias [38] introduced Anyplace, a navigation architecture that allows easy installation of additional modules, either for adding new features or extending its capabilities. It was openly distributed under MIT Licence, that allowed the Anyplace service to have more than 100,000 real user interactions by 2017.

Zafari et al. [39] provided a survey of indoor location systems (ILS) where the components of ILS are analysed. Thus, the authors discuss the relevance of ILS in multiple areas (healthcare systems, IoT, security, etc.), technologies, techniques and current challenges for ILS. One of the challenges mentioned in this article is standardization which is highly relevant to integrate different systems and improve the interconnection between components in modular systems.

An approach that deals with the heterogeneity of systems has been proved useful in the more generic field of distributed Cyber-Physical Systems (CPSs) [40], but without a focus on the specific aspects related to LBSs. On the other hand, vertical solutions have been proposed addressing specific requirements (e.g., energy efficiency [41]) or scenarios (e.g., exhibition venues [42]).

Table 1 shows the components used in the proposed integrated architecture and the comparison with current solutions analysed in previous paragraphs. As we can observe, the proposed architecture shares most of the characteristics implemented or described in other indoor positioning/localization systems, and other features which are described in the following sections, such as the discovery process. Moreover, the provided comparison takes into account the use of standards, which are considered as a challenge for today's IPS/ILS [39].

Table 1
Proposed architecture vs. current architectures.

Author	Out.	In.	S. F.	Disc.	Loc.	Nav.	Track.	Map	POIs	LBS	Stand.
* Proposal	*√	√	√	√	√	√	√	√	√	√	√
Kolodziej [29]	√	√	√	✗	√	√	√	√	√	√	√
**Gratsias et al. [30]	✗	√	✗	✗	√	√	√	√	√	✗	✗
di Flora et al. [31]	√	√	√	√	√	√	√	√	√	✗	√
**Kjærgaard [32]	-	-	-	✗	√	✗	√	√	✗	✗	✗
Molina et al [11]	✗	√	√	✗	√	✗	✗	√	√	✗	✗
Zeinalipour-Yazti et al. [38]	√	√	√	✗	√	√	√	√	√	✗	√

Out.: Outdoor, In.: Indoor, S.F.: Sensor Fusion, Disc.: Discovery process, Loc.: Location Nav.: Navigation, Track.: Tracking, Stand.: Standardization. LBS (Booking, Tour, Shoppinglist, etc.). * Taxonomy, proposal (no solution developed). ** Proposed integrated architecture.

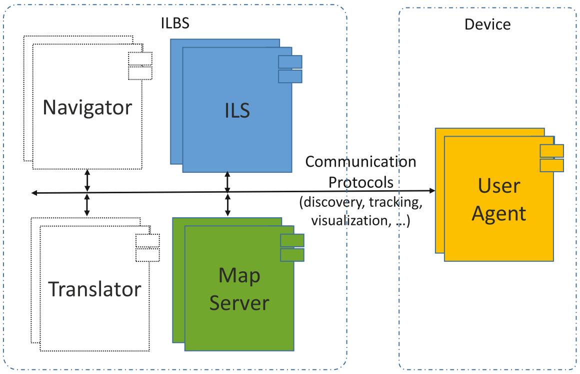


Fig. 1. The abstract LBS components.

It is essential to highlight that a few LBS do not provide information about the standards applied in their systems (see Table 1), and if those architectures can be integrated with other LBS. Furthermore, the services provided by these solutions are limited only to localization or positioning in some of the cases.

In this work, we propose an integrated solution that decomposes the systems offering LBSs into the functional abstract components that are most common in such systems, in order to achieve the goals of easing the processes of discovery, abstraction, composition, navigation, vertical and horizontal handover of positioning services.

Moreover, a very interesting paradigm, namely Multi-IoT based architectures [43], has been recently presented in the literature. It represents a step forward in the orchestration of multiple IoT systems in which, contrary to the classic IoT paradigm, a node is an instance of an IoT system and not only a generic IoT object. Therefore, in a Multi-IoT based architecture, multiple nodes each with different profile and behaviours could represent an object and its instance. The Multi-IoT paradigm investigates and implements IoT using the social network paradigm to capture the growing complexity of the IoT scenario. In that sense, Multi-IoT architectures offer the possibility to handle social aspects of IoT [44,45]; this could be particularly useful in our context, in which the LBS is correctly instanced multiple times with a proper profiling related to a specific context and user.

3. An integrated architecture for ILSs

A standardization effort to define common interfaces, protocols, and workflows among the components involved in the provisioning of LBSs is very important to reach a seamless transition between outdoor and indoor navigation, or while moving inside areas covered by different ILSs. In order to ease the standardization process, we grouped such components into six functional components: *ILS, Map Server, Navigator, Translator, User Agent* and *Communication Protocols* (see Fig. 1). Here we focus only on three of them, more details can be found in Furfari et al. [46].

The **Indoor Localization System (ILS)** consists of hardware and software components in charge of locating and tracking an entity (object or person), including possible components installed on person's device (e.g. radio frequency interfaces) and the environment (e.g. radio frequency access points). The **User Agent (UA)** is the abstract component that interacts with the ILS on behalf of the user, taking on the heterogeneity of the available systems. It is a component running on the entity's device, typically a smartphone. The metaphor that we consider most appropriate for the user interface of this component is that of instant messaging chats with which people are used to interact. The **Map Server** provides maps and associated context information such as POIs (Points of Interests). Map servers can be deployed locally either in the indoor area, or in the Cloud. Maps can be used to render graphical information as well as to calculate paths between a source and a destination within the coverage area.

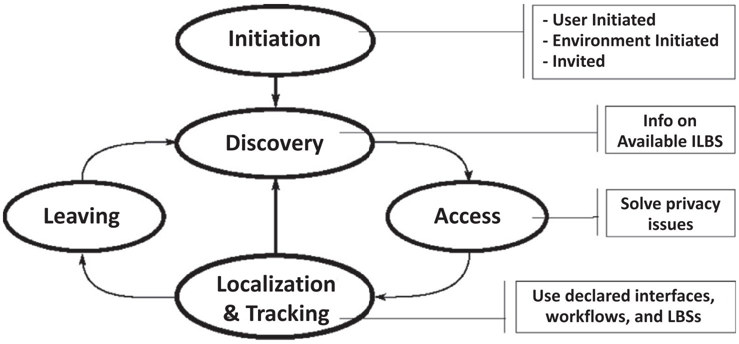


Fig. 2. Indoor navigation life-cycle.

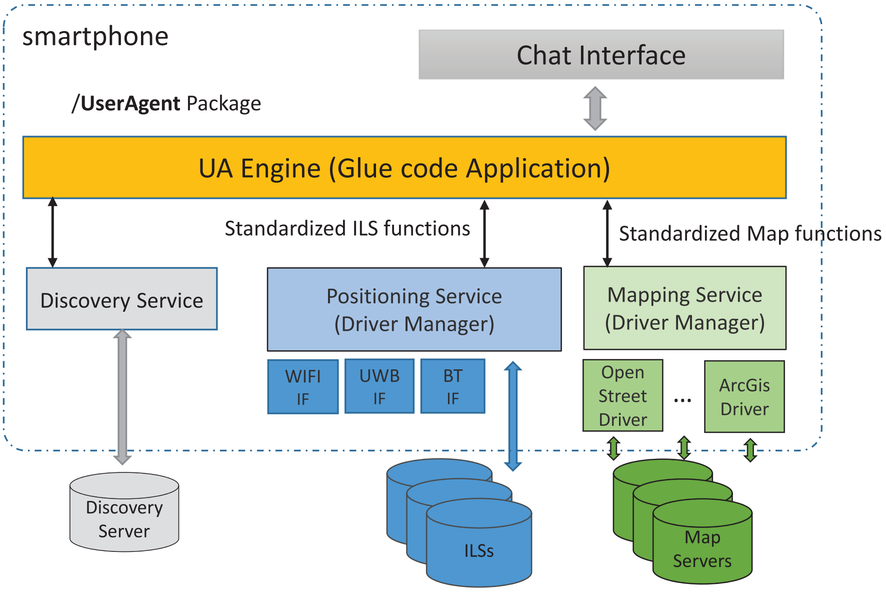


Fig. 3. User Agent abstract architecture.

The **Navigator** is the component that uses distance graphs to calculate routes to a specific target inside the coverage area, including special requirements such as avoiding stairs when carrying a load or supporting a group of people moving together. The **Translator** is the abstract component that performs a linguistic rendering of the information displayed on a map. As input tool the translator should facilitate the searching for a particular entity. Finally, with **Protocols** component we mean the set of standards used to allow interoperability of systems providing indoor LBSs. In general, standard agreement will be needed about:

- the discovery of localization systems and representation of their capabilities;
- description of workflows, for example how User Agent may cooperate with ILS in order to estimate its position;
- data streaming for tracking services;
- user interfaces for LBSs and navigation patterns; set of menus, icons to achieve common tasks.

The flow reported in Fig. 2 describes the various states of a **User Agent** during the interaction with the Infrastructure providing LBSs (**ILBS**).

The *Initiation* state represents how the localization process can be launched. The simplest case is when the user himself starts the process because inside an area where LBSs might be available. The *Discovery* state enables to search and retrieve the information relating to the available ILBSs. Different protocols can be used during the discovery phase, more specifically by using a local or global search, as reported in Section 4. Explicit notice and consent from the user should be exchanged during the system *Access* state. During the *Localization and Tracking* state the user is localized by the system, while the *Leaving* state is the abandonment of the localization service.

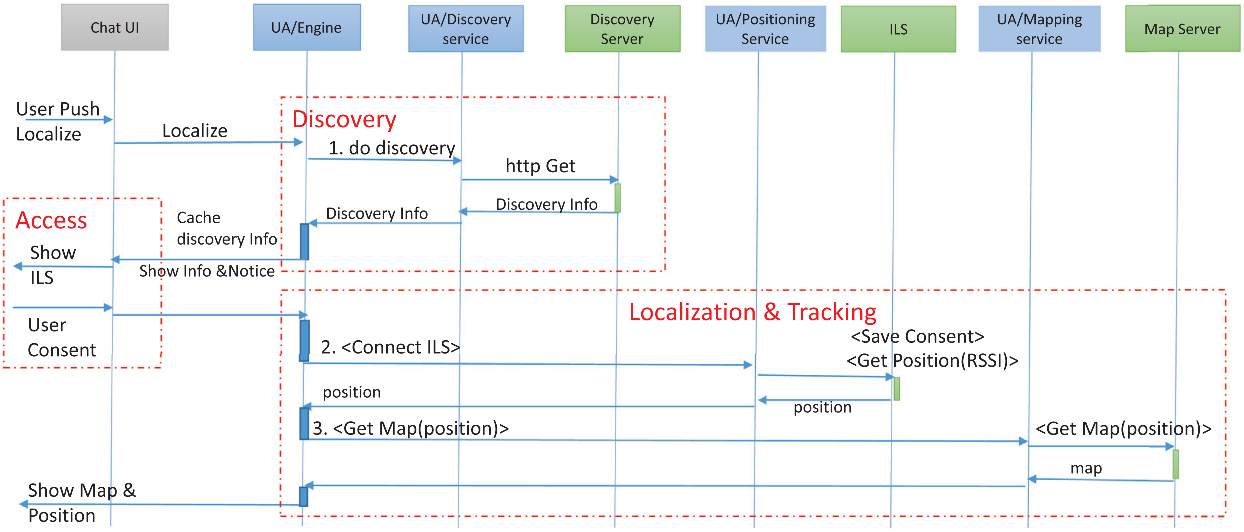


Fig. 4. Sequence diagram of the ILS discovery.

The Navigation life-cycle is mainly implemented by the User Agent (UA), the component running on the user's device. The UA abstract architecture (Fig. 3) is decomposed in components or services acting as the counterpart of Discovery server, ILS and Map server to which is needed to connect to.

We used the same schema of the Open DataBase Connectivity (ODBC) standard to represent connectivity of the User Agent with the rest of components. ODBC standard is based on driver providers that allow to connect to custom Data Sources. Because data sources can be different from each other (in some cases, very different), you need a way to translate standard ODBC function calls into the native language of each data source. Translation is the job of the drivers. Each driver accepts function calls through the standard ODBC interface and then translates them into code that is understandable to its associated data source. A similar ODBC schema can be used to connect to different type of ILSs and Map Servers. The User Agent business logic is implemented by the UA Engine; it is in charge of orchestrating all the communication with the various components. The UA engine represents the application layer. It is designed to use three services: the discovery, positioning and mapping services. Positioning and Mapping services are the corresponding of the ODBC Driver Manager a broker between the application layer and the installed drivers.

As an example, in Fig. 3 we included WiFi, UWB, and BT drivers to estimate positions and OpenStreet or ArcGIS drivers to load indoor maps. At the moment there is not a recognised standard for interfacing with the many technological solutions available for localization systems or map servers, thus in our prototype a practical approach has been used and the concrete UA architecture is slightly different.

The sequence diagram represented in Fig. 4 shows the interactions between the UA and the rest of the components highlighting the navigation life-cycle. The discovery process is initiated by the user and executed by the UA engine. Once retrieved the metadata describing the LBS infrastructure, it caches the info and notifies the user to get the consent to be tracked (*Access* phase in the figure). Without loss of generality, if the ILS is a remote positioning system, the UA engine, using the discovered info (*Discovery* phase in the figure), configures the positioning service to properly connect with the ILS (i.e., listening for RSSIs in WiFi-based systems). Then it sends the user consent and RSSI values to the ILS to receive an estimate position. Once received the position the UA engine requests the portion of map related to the user position and shows it (*Localization & Tracking* phase in the figure).

4. Discovery of location based services

The discovery phase represents a preliminary step in order to get access to an ILBS. This step is also reported with the sequence diagram in Fig. 4: the Chat UI firstly localizes the ILBS with the discovery phase (marked as a red-dashed line). Similarly to other networked resources (e.g., printers, fax or shared folders), we consider that an ILBS can be described with a set of meta-information structured according to an open data description format. Such information can be first discovered, and then they can be used by an User Agent for the configuration and localization purposes.

We argue that the meta-information characterizing the ILBS has to be standardized so that every User Agent can interact with ILBS from different vendors. To this purpose, the features of a generic ILBS can be modelled with taxonomy, we refer to [46] for a preliminary discussion of such standardization effort.

We report in the following an example of meta-information, represented with a tree structure rooted on the ILBS node:

▼ ILBS

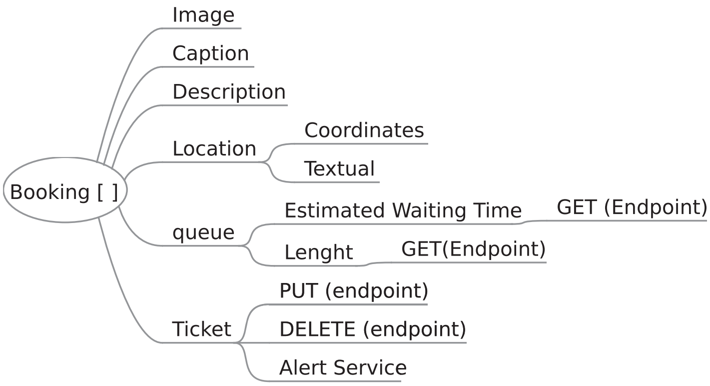


Fig. 5. Details of a LBS node.

▶ Identification
▶ Authority
▼ ILS
  Type : RF/WiFi/RemoteProcessing@ILS-Taxonomy
  EndPoint : <value>
▶ MapServer
▶ Navigator
▶ Translator
▶ GeoFence
▶ System-Agent
▼ LBS
  ▶ Tour
  ▶ Booking
  ▶ ShoppingList

We propose a set of sibling nodes each of them describes a feature of the ILBS. Among them we cite:

- *Identification*: a unique URI pointing to the ILBS. The URI identifies the ILBS resource on the network, and it represents the initial end-point providing some basic APIs for localizing the user's device.
- *Authority*: it groups together a set of information related to privacy and security issues, for example provides information on the system's accountability: a responsible and the procedures to follow to solve any privacy concern as well as certificates used for authentication.
- *ILS*: it provides the URIs to access the ILS interfaces and describes the system features: technique (Time of Arrival, Angle of Arrival, Received Signal Strength, etc.), technology (WiFi, BT, UWB, etc.), accuracy and so on.
- *System Agent*: an intelligent process that impersonates an area supervisor and offers advanced services. Using the instant messaging metaphor, the system-agent can be implemented as a chat user with whom the person can talk and request information: the position of an item, the path to reach a zone, or even the available services just described. In Section 5, we describe a prototype based on Telegram client that instantiates a system-agent chat dynamically.
- *Map Server*: it reports the URI from which fetching the indoor maps. The Map Server can provide the indoor map according to several formats: vectorial map, tiled map with different zoom levels.
- *GeoFence*: it describes the indoor area covered with this ILBS. The area can be described according to different formats: coordinates of the bounding box (lower left and upper right), WKT coordinates, KML markup, etc.
- *LBS*: this node lists the services that ILBS offers to the User Agent. Examples of services are: Tour for planning a tour indoor, Booking for reserving a ticket, Shopping List for organizing the path of the products to pick in a shopping mall, etc.

As a representative example, we further detail an LBS provided by the ILBS. An LBS is a service exploiting the knowledge of the location of the end-users. An example of LBS is the Booking service, a modern version of the ticket dispenser service, that can be used to easily book a service and to notify the end-user it is time to access the service. The Booking service is useful to limit the waiting queue in front of a service and, in turn, to limit crowded areas indoors. The Booking service offered by the ILBS can be further detailed with the information reported in Fig. 5.

4.1. Local and global search

We now describe how the meta-information of an ILBS can be obtained. We consider two possible approaches: local and global discovery. The local discovery allows the identification of an ILBS in the nearby, e.g. in a range of a few meters from the User Agent. To this purpose, short-range network interfaces such as Bluetooth [47], WiFi Direct or WiFi probes [48] can be used. Such interfaces allow to broadcast a short notification periodically, the information carried with the notification can also include the URI to the ILBS's meta-information. As a meaningful example, we cite the EddyStone payload of the Bluetooth Low Energy protocol. Such payload has been designed to carry an URL, hence the ILBS can advertise its URL and enable a User Agent to be notified and to retrieve the meta-information as regular text-based content. The coverage radius of the aforementioned network interfaces varies indoor (e.g. 10–20 m) according to the existence of obstacles in between the emitter and the receiver, the existence of network interference and the position of the emitting device (e.g smartphone of the user placed on the back pocket).

Differently, the global discovery requires the User Agent to access the Internet in order to retrieve the ILBS meta-information. The global discovery reproduces the way the people browse the Web: people query a search engine in order to fetch the street address or the phone number of a service. Similarly, we consider that it is also possible to query a search engine in order to obtain the URI of the ILBS, so that to reproduce the user-experience of the local search. Our vision is that in the next future, the search engines will also provide to the user the URI of the ILBS covering the place searched, similarly to the applications launched with the retrieved street address or phone number information, such URI, thanks to a specific scheme, will trigger the launch of applications (i.e. User-Agent Interface) that automatically connects to the local ILS and guides the user in the indoor environment. Global search can be used even remotely, that is when the user is not physically present in the area covered by the ILS and other interesting services, for example to know how crowded the environment is, can be used.

Local and global discovery can coexist together. The User Agent can try to look for nearby ILBS with the short-range network interfaces and, eventually, switch to a global search if no ILBS are locally discovered.

5. Prototyping the integrated solution

In order to provide an implementation of the concepts described in this article, we developed *LOCgram*: a modified version of the popular messaging app Telegram. Telegram for Android is open source and freely available.[2]

The proposed work aims to propose a unified approach for discovering and interacting with different indoor localization systems deployed into the same indoor environments. Such systems are composed by three fundamental components (or subsystems), namely ILS (i.e., the unit able to estimate the user's position), Map Server (i.e., the unit which dynamically provides indoor maps) and User Agent (i.e., the unit which requires and consumes positioning information).

LOCgram extends the Telegram interface with localization functionalities and implements the User Agent business logic.

As previously described, positioning and mapping services require a further standardization effort that we foresee in a close future.

Therefore, the User Agent implementation is similar to the abstract architecture proposed in the Fig. 3 but makes use of mapping libraries that partly perform the task of abstracting from the various Map Servers available today.

By interacting with the LOCgram interface proposed herein, and following the sequence diagram shown in Fig. 4, a user may start a localization process obtaining information about possible localization systems and services available into the environment. Through the meta-information obtained, the application can guarantee a proper access phase to the localization and tracking capabilities. Furthermore, in the same application, the user obtains a dynamic rendering of the indoor map.

We customized the source code with a new menu item *localize* to the side panel, as shown on the left of Fig. 6. This will show the indoor Localization Panel through which we can control the localization procedure. The middle part of Fig. 6 shows a preliminary version of the indoor localization panel. Pressing button "START LOCALIZATION" activates the localization procedure which is carried out by an Android service named *LocalizationEngine*.

In this preliminary version of LOCgram the goal of the LocalizationEngine service is to:

1. Discover the ILBS's entry-point (see Section 4) and download the JSON file containing such information;
2. Configure the User Agent with discovered information;
3. Rendering the local map provided by a Map Server.

The rest of this section details, for each of the three steps, some implementation aspects characterizing our prototype.

5.1. ILBS discovery with a local search

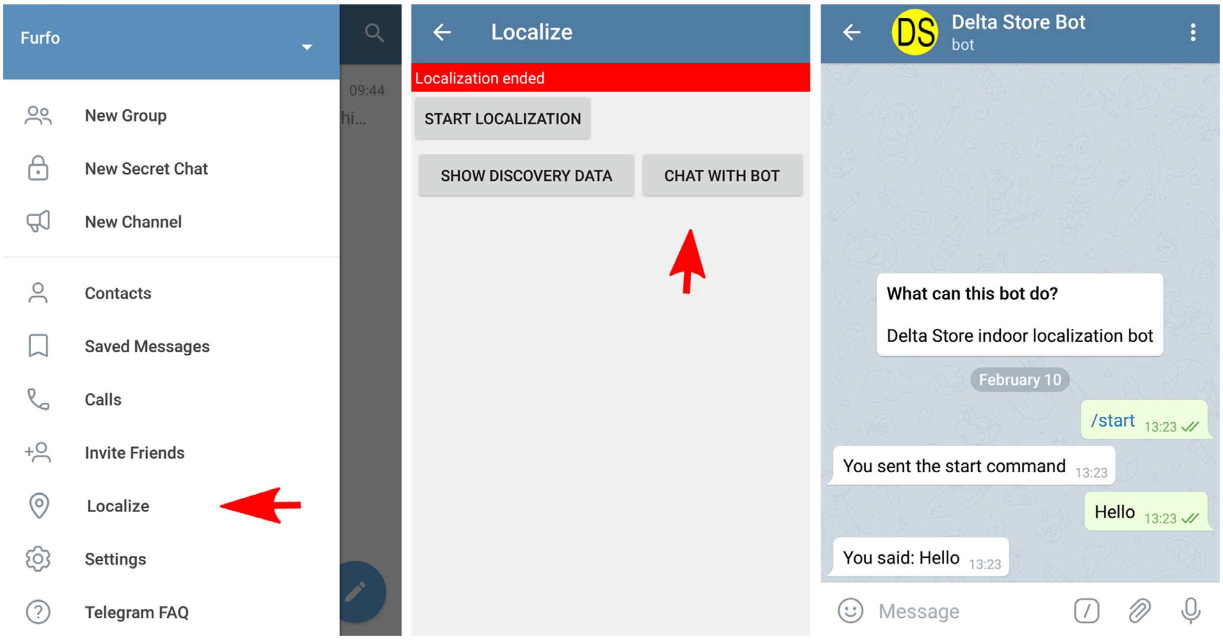
We implement the ILBS's discovery phase implementing a service named *DiscoveryService* that coordinates other services each taking care of a specific discovery technology. As discussed in Section 4, we consider two searches: local and global. For the purpose of this work, we only focus on a local search implemented with Beacons Bluetooth (BLE) and WiFi probing.

Service *BeaconDiscoveryService* performs a timed Bluetooth scan, looking for EddyStone beacons.[3] Such kind of Bluetooth's beacon can encode a URL inside the payload. Upon receiving beacon results, the DiscoveryService implements a ranking policy in order to select the closest beacon. More specifically, the DiscoveryService collects all the beacons for a time period

---

[2] https://github.com/DrKLO/Telegram

[3] https://developers.google.com/beacons/eddystone



Localization ended

What can this bot do?

Delta Store indoor localization bot

February 10

/start 13:23

You sent the start command 13:23

Hello 13:23

You said: Hello 13:23

Fig. 6. Examples of LOCgram: ILBS discovery and interaction with the Bot.

and then it averages the RSSI (Received Signal Strength Indicator) values for each beacon and it sorts the contained URLs in a decreasing order (from the highest to the lowest RSSI average value). Such policy allows to select the reliable beacon with respect to the user agent, so that to implement a local-based selection policy. After selecting the proper beacon, the DiscoveryService tries to retrieve ILBS entry-point by fetching the URLs carried with the beacons.

We also implemented a WiFi-based local search. It is implemented with the *WiFiDiscoveryService*. It takes care of an alternative discovery procedure based on WiFi SSIDs. The service performs a WiFi scans looking for SSIDs starting with the prefix *ILBS-* and it tries to decode the characters following the prefix as a URL (e.g., ILBS-https://tiny.cc/ipakjz is an example of a valid SSID). Similarly to the selection policy described for the BeaconDiscoveryService, also in this case the detected SSID are ranked basing on RSS values.

After the local search provides at least a candidate URL, the user agent tests the URLs discovered. This procedure verifies if the entry-point received during the discovery phase strictly concern a LBS. Indeed, the procedure first sends an HTTP HEAD request to the target URL with header “Accept: application/vnd.isti.ilbs+json”. The custom MIME type is meant to discard sites that do not offer ILBS discovery data. Only if the HTTP response contains a “Content-type: application/vnd.isti.ilbs+json” header, the URL can respond to a GET HTTP request to retrieve the actual JSON document containing ILBS data. Using HEAD before GET allows us to retrieve just the headers and verify that the URL is acceptable before actually downloading a potentially large response. We also allow for a (single) redirect (3xx) response in order to handle URL compression schemes like Bitly or TinyURL.

5.2. ILBS discovery with a global search

The purpose of the *global search* is for a user to bypass the discovery procedure and retrieve the url of the ILBS server through the web page of a store/organization that offers localization services. The integration with LOCgram can be achieved thanks to Android implicit intents and intent filters, as well as the way that common Android browsers work when presented with a url with a scheme they cannot handle.

When an app starts an activity with an implicit intent, Android tries to match it against existing activities for any apps on the device and starts a suitable activity if it finds one. An implicit intent can contain various information including an action, a category, a url (data) and parameters (extras). The AndroidManifest.xml file for LOCgram was modified to associate a new intent filter with the main activity *LaunchActivity*, as follows.

```
<activity android:name=’’org.telegram.ui.LaunchActivity’’ \del{...}\ins{\ldots}  >
...
<intent-filter>
<action android:name=’’android.intent.action.VIEW’’ />
<category android:name=’’android.intent.category.DEFAULT’’ />
<category android:name=’’android.intent.category.BROWSABLE’’ />
<data android:host=’’localhost’’ android:scheme=’’ilbs’’ />
```

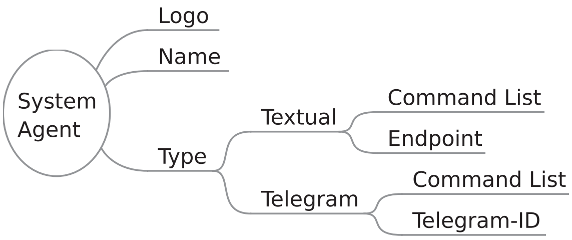


Fig. 7. Details of the system-agent node.

Table 2
Existing solutions for map retrieval and rendering.

	Map-view sol.	Dev.	Custom.	Map Ope.	D.S.	Licensing	Offline	Coord.	Support
Native	ArcGIS	3rd-party	High	All	L7	Proprietary	Yes	Yes	Low
	Google	3rd-party	Low	All	L4	Proprietary	Yes	Yes	Low
	MapBox	3rd-party	Low	All	L7	Proprietary	Yes	Yes	Low
	Image Viewer^{+}	Own	High	Pan, Zoom	L2	Open Source	Yes	No	Low
	Tile Viewer^{+}	Own	High	Pan, Zoom	L3	Open Source	Yes	No	Low
Web	ArcGIS	3rd-party	High	All	L7	Proprietary	No	Yes	Low
	Google	3rd-party	High	All	L4	Proprietary	No	Yes	Low
	MapBox	3rd-party	High	All	L7	Proprietary	No	Yes	Low
	OpenLayers	3rd-party	High	All	L7	Open Source	Partial	Yes	Medium
	Leaflet	3rd-party	High	All	L7	Open Source	Partial	Yes	High

“All” stands for: pan, zoom, pop-up and layer selection.

```
</intent-filter>
...
</activity>
```

The `<intent-filter>` xml element advertises that activity *LaunchActivity* is willing to receive and handle intents that match certain criteria. The `<category>` xml element declares that this activity can be used to display data referenced by a link in a browser. The `<data>` xml element requires that acceptable link urls must have “ilbs” as a schema and “localhost” as host. If the web page of the store contains a link like:

<a href=”ilbs://localhost?url=http://ilbs.example.com/ilbs”>

`localize me`

`</a>` and a user clicks this link, an Android browser (being unable to manage the `ilbs://` scheme itself) will try to start an activity able to visualize what is referenced by that link. Android will consider *LaunchActivity* as a matching Activity and will pass it parameter url=http://ilbs.example.com/ilbs as an intent extra. Upon receiving this intent, LOCgram will bypass the discovery procedure with bluetooth/wifi and proceed to directly download ILBS data from the url parameter.

5.3. Configuring the user agent

Configuring the User Agent consists of extracting the value of specific fields in the JSON document (meta-information describing the LBS Infrastructure) and passing them to the internal UA services: Positioning and Mapping service. We tested even the dynamic creation of a *system-agent chat* by checking, through Telegarm APIs, that the system-agent field value available in the JSON document corresponds to a valid Telegram Bot id. In Fig. 7 the meta-information related to the system-agent are exploded. The Localization Panel will then enable button “CHAT WITH BOT” (see the middle part of Fig. 6). When pressed this button will open a Telegram chat with the Bot discovered in the previous steps and will allow the user to send commands and request information.

5.4. Rendering indoor maps

The maps shown in modern apps are mainly dynamic maps (e.g., Google Maps and OSM) that change the information they render (visualize) according to the user interaction. Commonly, those maps contain a basemap, a set of data layers, an extent, and navigation tools to pan and zoom. Everything shown in a dynamic map belongs to a layer (of information). That organization into layers allows to hide or show map information as, for example, a result of a request of the user to show a specific building floor, or a result from zooming in.

To address the map rendering, we explored mapping solutions that allow actions related to map visualization and interaction, like displaying an indoor position estimation, that basic LBS may require in a smartphone’s app. The solutions are grouped in two categories: those implemented in native (Android and iOS) code, and those implemented using Web technologies (i.e., HTML, CSS, javascript), as presented in Table 2. The Web solutions consist of a Web page that is rendered in

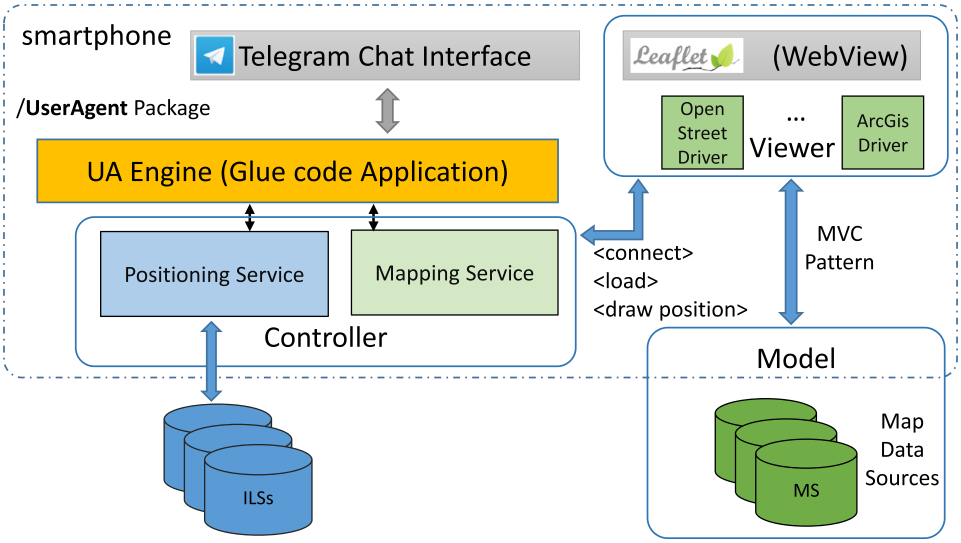


Fig. 8. Map loading and map interaction.

a Web component in the native application, being the native code able to interact with the Web solution. The table shows libraries widely popular for creating mobile and web map-based applications and two modest options that may be implemented and released for their open usage. In the later options, the plus (+) sign highlights that they include basic simple vector shapes drawing capacity, which is mandatory for our goal actions. The solutions are described in Table 2 using 8 dimensions, namely developer, customization, map operations, data source, licensing, off-line maps, coordinates transformation and support from community.

The data source dimension comprises 7 values, from **L1** to **L7**, of increasing and cumulative levels of support. **L1** solutions can only display static images. **L2** solutions can also draw simple vector shapes, like markers and lines. **L3** solutions may additionally create a map by composing tiles fetched from a predefined service. **L4** solutions include widely available basemap(s), likely vendor specific. The **L5** ones also support layers created from OGC compliant map services. **L6** solutions can create layers from map services from a specific vendor. Finally, the **L7** ones can include layers from map services from several vendors. The data source value of a solution may be boosted up if an intermediate translation service is added.

The 3rd party native solutions explored here are excellent for building map-based mobile applications and support off-line maps. However, apart from ArcGIS, which may consume online-defined Web Maps, it is more difficult to change the map in native solutions than in the Web ones. The own implemented solutions, while eliminate possible dependency-breakdowns, suffer from low data source support, poor user interaction, and low mapping capabilities. All explored Web-based solutions are 3rd party and provide platform-independency, easy remote configurations, and rich user interactions and mapping capabilities. However, the OpenLayers and Leaflet solutions stand out because of a partial offline maps support. In particular, we chose the Leaflet-based solution as our proposal because its high community support (in the form of plugins) allows outstanding capabilities of supported map operations and data sources.

Fig. 8 shows a proposed solution for map rendering and interaction in the context of the abstract architecture. The Mapping service interacts with a component (WebView) that provides Web browser capabilities, to visualize map-related information as requested by other services. For example, information drawn from the Positioning Service may be supplied to the Mapping Service, which contains at least the URL where the map is published. The Mapping Service instructs the WebView to load the web page that displays the map. As the page loads, the map is rendered by a Web mapping library (Leaflet), which also handles data fetching from distinct map data sources. After the map is ready, the Mapping Service may receive new commands, (e.g., to update a position estimate marker or to trace a route), which it translates into requests to the execution of predefined functions in the web page.

5.5. Measuring the integration of multiple ILSs

We finally discuss in this subsection how to measure the performance the LBS architecture defined in Section 3. In particular, we focus on identifying some KPIs (Key Performance Indicator) meant to improve the final user-experience. Our long-term vision aims at supporting the implementation of indoor services with performance similar to the current outdoor services available on our smart-devices (e.g. Google Maps, GPS sport trackers, geo-fancing apps etc). Such a process is not an easy task at all, since indoor environments introduce more challenging requirements for a LBS architecture.

We identify 4 KPIs, as follows:

- *Discovery Latency*: this KPI measures the time interval between a user entering in a monitored environment and the time the system discovers and binds to one of the available ILS. The lower the latency, the quicker the user can be successfully localized and hence he/she can access the indoor services available.
- *Initial Location*: this KPI measures the latency in computing the initial position of a user. The ILS might introduce a certain delay in computing the initial position, as the user keeps moving and its device provides new data to be elaborated by the ILS.
- *Handover overhead*: this KPI measures the overhead introduced by the system in order to leave an ILS and to discover and to bind to a different one. The handover represents a challenging feature since the system has to switch the ILS in a seamless way, without providing any evidence of a system lock. In particular, we identify two critical handover scenarios that might negatively affect this KPI:
    - *outdoor-indoor transition*: the user moves from outdoor to an indoor environment, switching from a GPS positioning system to an indoor positioning system;
    - *out-of-signal*: the user moves from an uncovered to a covered indoor environment and vice-versa. In this scenario, the user's device might intermittently provide data to the ILS.
- *Updating interval*: this KPI refers to the capability of the ILS to refresh periodically the user's position. The updating interval resembles the keep-alive mechanisms adopted by many network protocols. The user has to receive periodically its position so that to feel the perception the ILS keeps working properly. Deadlock situations in which the user position is outdated bring the user to stop/restart or kill the apps, all behaviour that needs to be mitigated.

Measuring such KPIs represents a challenging task, since it is required to measure quantitatively metrics at different layers of the LBS architecture we propose. Besides, the architectural design can impact KPIs: for example the discovery process could notify the user-agent of all the ILSs available in a big area in one shot, as well as provide meta-information, like geo-fences, to speed up the handover between different ILSs. All such aspects and their possible solutions have to be addressed and analyzed through on-site testing; indoor competitions like EvAAL [49,50], now in its ninth edition, is a suitable international event to launch a new track about integration and interoperability of multiple ILSs.

6. Conclusion

Due to the use of different technologies such as WiFi, 5G, Bluetooth, UWB, RFID, etc. on a wide scale, we believe that future networks will be highly heterogeneous. Therefore, it is highly likely that a localization system will be a hybrid system that might rely on a number of technologies. To obtain improved performance and a seamless user experience, there might be the need for a common and shared architecture and taxonomy. In this paper, we propose an integrated architecture for indoor localization systems. In particular, we define a service discovery method and we identified a list of KPIs that we consider critical for an efficient integration of multiple Indoor Localization Systems. In order to validate the proposed solution, we also implement the proposed integrated architecture by focusing on the development of a User Agent based on Telegram and by developing the proposed discovery phase to retrieve map, services, and systems information. This topic represents a key factor in the extensive adoption of integrated LBSs and future works will focus on the scalability of the proposed approach, in terms of a number of managed ILSs to be deployed in a wide area, and its implication on the goals identified in the paper as easing the process of discovery, abstraction, composition, navigation, vertical and horizontal handover of available positioning services in an area.

As a future work, we plan to realize a prototype of the architecture proposed able to serve indoor positioning information in public environments such as research campuses or hospitals. Furthermore, we plan to integrate our proposal into the Multi-IoT paradigm, in which the proposed LBS becomes an object instance into a multiple IoT architecture that handles the social aspects of the user's context.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge funding from the European Union's H2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No. 813278 (A-WEAR, http://www.a-wear.eu/).

References

[1] D. Macagnano, G. Destino, G. Abreu, Indoor positioning: a key enabling technology for IoTapplications, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), IEEE, 2014, pp. 117–118.
[2] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805.
[3] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, O. Rana, The internet of things, fog and cloud continuum: integration and challenges, Internet Things 3 (2018) 134–155.
[4] I.A. Junglas, R.T. Watson, Location-based services, Commun. ACM 51 (3) (2008) 65–69.

[5] B. Khoo, RFID- from tracking to the internet of things: areview of developments, in: 2010 IEEE/ACM Int'l Conference on Green Computing and Communications Int'l Conference on Cyber, Physical and Social Computing, 2010, pp. 533–538.
[6] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, G. Marrocco, RFID technology for iot-based personal healthcare in smart spaces, IEEE Internet Things J. 1 (2) (2014) 144–152.
[7] A. Hameed, H.A. Ahmed, Survey on indoor positioning applications based on different technologies, in: 2018 12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), IEEE, 2018, pp. 1–5.
[8] infsoft, White paper: Indoor Positioning & Services, Technical Report EN 2019-7, infsoft GmbH, 2019. (accessed April 28, 2020) URL: https://www.infsoft.com/Portals/0/Images/solutions/basics/whitepaper/infsoft-Whitepaper-EN-IndoorPositioning_download.pdf
[9] U. Rueppel, K.M. Stuebbe, Bim-based indoor-emergency-navigation-system for complex buildings, Tsinghua Sci. Technol. 13 (S1) (2008) 362–367.
[10] A. Al Rifai, Indoor Positioning at Arlanda Airport, KTH, Communication Systems, CoS, 2009 Master's thesis.
[11] B. Molina, E. Olivares, C.E. Palau, M. Esteve, A multimodal fingerprint-based indoor positioning system for airports, IEEE Access 6 (2018) 10092–10106.
[12] T.J. Gallagher, B. Li, A.G. Dempster, C. Rizos, A sector-based campus-wide indoor positioning system, in: 2010 International Conference on Indoor Positioning and Indoor Navigation, 2010, pp. 1–8.
[13] J. Torres-Sospedra, J. Avariento, D. Rambla, R. Montoliu, S. Casteleyn, M. Benedito-Bordonau, M. Gould, J. Huerta, Enhancing integrated indoor/outdoor mobility in a smart campus, Int. J. Geograph. Inf. Sci. 29 (11) (2015) 1955–1968, doi:10.1080/13658816.2015.1049541.
[14] H. Hakim, R. Renouf, J. Enderle, Passive RFID asset monitoring system in hospital environments, in: Proceedings of the IEEE 32nd Annual Northeast Bioengineering Conference, 2006, pp. 217–218.
[15] L. Calderoni, M. Ferrara, A. Franco, D. Maio, Indoor localization in a hospital environment using random forest classifiers, Expert Syst. Appl. 42 (1) (2015) 125–134, doi:10.1016/j.eswa.2014.07.042.
[16] I. Rubino, J. Xhembulla, A. Martina, A. Bottino, G. Malnati, Musa: using indoor positioning and navigation to enhance cultural experiences in a museum, Sensors 13 (12) (2013) 17445–17471, doi:10.3390/s131217445.
[17] J.D.G. Gallardo, F. Aranda-Polo, T.A. Benítez, F.Á. Franco, LIMUS: exploration of technological prototypes for location-based services in museums, in: F. Potortì, V. Renaudin, K. O'Keefe, F. Palumbo (Eds.), Short Paper Proceedings of the Tenth International Conference on Indoor Positioning and Indoor Navigation - Work-in-Progress Papers (IPIN-WiP 2019) co-located with the Tenth International Conference on Indoor Positioning and Indoor Navigation (IPIN 2019), Pisa, Italy, September 30th - October 3rd, 2019, CEUR Workshop Proceedings, 2498, CEUR-WS.org, 2019, pp. 416–422.
[18] P. Bellavista, A. Küpper, S. Helal, Location-based services: back to the future, IEEE Pervasive Comput. 7 (2) (2008) 85–89.
[19] M.F.S. van der Ham, S. Zlatanova, E. Verbree, R. Voûte, Real time localization of assets in hospitals using Quuppa indoor positioning technology, ISPRS Ann. Photogramm. Remote Sens.Spat. Inf. Sci. IV-4/W1 (2016) 105–110, doi:10.5194/isprs-annals-IV-4-W1-105-2016.
[20] R. Brena, J. García-Vázquez, C. Galván Tejada, D. Munoz, C. Vargas-Rosales, J. Fangmeyer Jr, A. Palma, Evolution of indoor positioning technologies: asurvey, J. Sens. 2017 (2017), doi:10.1155/2017/2630413.
[21] S. Husain, A. Prasad, A. Kunz, A. Papageorgiou, J. Song, Recent trends in standards related to the internet of things and machine-to-machine communications, J. Inf. Commun. Converg. Eng. 12 (4) (2014) 228–236.
[22] Z. Sheng, S. Yang, Y. Yu, A.V. Vasilakos, J.A. McCann, K.K. Leung, A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities, IEEE Wirel. Commun. 20 (6) (2013) 91–98.
[23] G. Wu, S. Talwar, K. Johnsson, N. Himayat, K.D. Johnson, M2M: from mobile to embedded internet, IEEE Commun. Mag. 49 (4) (2011) 36–43.
[24] J. Song, A. Kunz, M. Schmidt, P. Szczytowski, Connecting and managing M2M devices in the future internet, Mob. Netw. Appl. 19 (1) (2014) 4–17.
[25] J. Kim, J. Yun, S.-C. Choi, D.N. Seed, G. Lu, M. Bauer, A. Al-Hezmi, K. Campowsky, J. Song, Standard-based IoT platforms interworking: implementation, experiences, and lessons learned, IEEE Commun. Mag. 54 (7) (2016) 48–54.
[26] P. Fonseka, K. Sandrasegaran, Indoor localization for IoT applications using fingerprinting, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), IEEE, 2018, pp. 736–741.
[27] Z. Turgut, G.Z.G. Aydin, A. Sertbas, Indoor localization techniques for smart building environment, Procedia Comput. Sci. 83 (2016) 1176–1181.
[28] D. Mocrii, Y. Chen, P. Musilek, IoT-based smart homes: a review of system architecture, software, communications, privacy and security, Internet Things 1 (2018) 81–98.
[29] K.W. Kolodziej, OpenLS for indoor positioning: strategies for standardizing location based services for indoor use, Massachusetts Institute of Technology, 2004 Ph.D. thesis.
[30] K. Gratsias, E. Frentzos, V. Delis, Y. Theodoridis, Towards a taxonomy of location based services, in: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 19–30.
[31] C. di Flora, M. Ficco, S. Russo, V. Vecchio, Indoor and outdoor location based services for portable wireless devices, in: 25th IEEE International Conference on Distributed Computing Systems Workshops, 2005, pp. 244–250.
[32] M.B. Kjærgaard, A taxonomy for radio location fingerprinting, in: J. Hightower, B. Schiele, T. Strang (Eds.), Location- and Context-Awareness, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 139–156.
[33] L. Wirola, T.A. Laine, J. Syrjärinne, Mass-market requirements for indoor positioning and indoor navigation, in: 2010 International Conference on Indoor Positioning and Indoor Navigation, 2010, pp. 1–7.
[34] T. Ruiz-López, J.L. Garrido, K. Benghazi, L. Chung, A survey on indoor positioning systems: foreseeing a quality design, in: A.P. de Leon F. de Carvalho, S. Rodríguez-González, J.F. De Paz Santana, J.M.C. Rodríguez (Eds.), Distributed Computing and Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 373–380.
[35] F. Lemic, V. Handziski, N. Mor, J. Rabaey, J. Wawrzynek, A. Wolisz, Toward standardized localization service, in: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2016, pp. 1–8.
[36] G. Stevenson, J. Ye, S. Dobson, P. Nixon, Loc8: a location model and extensible framework for programming with location, IEEE Pervasive Comput. 9 (1) (2009) 28–37.
[37] T. Graichen, E. Gruschka, U. Heinkel, A map framework using crowd-sourced data for indoor positioning and navigation, in: 2017 IEEE International Workshop on Measurement and Networking (M N), 2017, pp. 1–6, doi:10.1109/IWMN.2017.8078395.
[38] D. Zeinalipour-Yazti, C. Laoudias, The anatomy of the anyplace indoor navigation service, SIGSPATIAL Spec. 9 (2) (2017) 3–10, doi:10.1145/3151123.3151125.
[39] F. Zafari, A. Gkelias, K.K. Leung, A survey of indoor localization systems and technologies, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2568–2599.
[40] G. Tanganelli, L. Cassano, A. Miele, C. Vallati, A methodology for the design and deployment of distributed cyber-physical systems for smart environments, Future Gener. Comput. Syst. 109 (2020) 420–430.
[41] P. Geraldo Filho, L.A. Villas, V.P. Gonçalves, G. Pessin, A.A. Loureiro, J. Ueyama, Energy-efficient smart home systems: infrastructure and decision-making process, Internet Things 5 (2019) 153–167.
[42] H. Li, J.K. Ng, V.C. Cheng, W.K. Cheung, Fast indoor localization for exhibition venues with calibrating heterogeneous mobile devices, Internet Things 3 (2018) 175–186.
[43] G. Baldassarre, P.L. Giudice, L. Musarella, D. Ursino, The miot paradigm: main features and an "ad-hoc" crawler, Future Gener. Comput. Syst. 92 (2019) 29–42.
[44] D. Ursino, L. Virgili, Humanizing IoT: defining the profile and the reliability of a thing in a multi-IoT scenario, in: Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications, Springer, 2020, pp. 51–76.
[45] F. Cauteruccio, L. Cinelli, G. Fortino, C. Savaglio, G. Terracina, D. Ursino, L. Virgili, An approach to compute the scope of a social object in a multi-IoT scenario, Pervasive Mob. Comput. 67 (2020) 101223.
[46] F. Furfari, A. Crivello, P. Barsocchi, F. Palumbo, F. Potortì, What is next for indoor localisation? Taxonomy, protocols, and patterns for advanced location based services, in: 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2019, pp. 1–8, doi:10.1109/IPIN.2019.8911759.

[47] P. Barsocchi, A. Crivello, M. Girolami, F. Mavilia, F. Palumbo, Occupancy detection by multi-power bluetooth low energy beaconing, in: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1–6, doi:10.1109/IPIN.2017.8115946.
[48] F. Potortì, A. Crivello, M. Girolami, E. Traficante, P. Barsocchi, Wi-Fi probes as digital crumbs for crowd localisation, in: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016, pp. 1–8, doi:10.1109/IPIN.2016.7743599.
[49] F. Potortì, A. Crivello, F. Palumbo, The Evaal Evaluation Framework and the Ipin Competitions, in: Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation, Elsevier, 2019, pp. 209–224.
[50] P. Barsocchi, S. Chessa, F. Furfari, F. Potorti, Evaluating ambient assisted living solutions: the localization competition, IEEE Pervasive Comput. 12 (4) (2013) 72–79.