Geolocation Architecture for Combining Multiple Technologies

Jean-Baptiste Drucker Tony Ducrocq Colombe Herault Guillaume Lefebvre Eric Ptak

Worldline, R&D User eXperience
Lille, France
Email: name.surname@worldline.com

Abstract—**Industry, retail and logistic markets are more and more asking for geolocation solutions, especially indoor ones. They often have different needs at the same time and require several different technologies. Until now, there was no solution for combining multiple technologies. To meet these needs, we propose a distributed, multi-user, multitechnology architecture for combining different geolocation solutions.**

Keywords–Indoor-location; middleware; distributed;

I. Introduction

Context-aware computing, in particular based on geolocation, is a big trend in the field of mobile computing. The market of location-based services is mainly driven by retailers, who want to improve their Customer Relationship Management by providing services such as couponing in the best place at the best time. But it is also boosted by the health market and Business to Business applications, such as those for maintenance and traceability uses.

Recently, cheap, easy-to-deploy technologies such as iBeacon [1] have been promoted, but lots of other technologies, such as Wi-Fi fingerprinting [2], can provide valuable solutions for geolocation. Nevertheless, the development of geolocation-based applications depends very much on the chosen technology. Therefore, it is not possible to use a unique application in several places if they are not equipped with the same location system. For example, if a retailer manages several shops, one that is equipped with iBeacon, another with inaudible sound, and the last one with Wi-Fi fingerprinting, then three different implementations of the localization application have to be developed.

Moreover, it is not easy to combine technologies to use them concurrently and provide enhanced services that dynamically take advantage of all the equipment present. For instance, in hospitals, parts of buildings, such as Computer Tomography scanner rooms, are equipped with accurate, highly available location systems that are based on active Radio Frequency Identification (RFID) [3] or Wireless-Fidelity (Wi-Fi) tags [4]. They aim at locating patients, healthcare equipment and patient records in order to improve medical furniture usage and reduce the number of medical errors. But these hardware systems (antennas and tags) are quite expensive and hard to deploy. Therefore, only critical rooms are equipped with them.

On the other hand, more and more patients and nursing staff own mobile phones that support Bluetooth Low Energy. Furthermore, deploying iBeacons that could be attached to trolleys or stretchers is quite easy and inexpensive. Therefore, it is fairly easy to implement an application that triggers an alert when a staff member moves away from their medical furniture.

However, today, no simple solution takes advantage of the fact that smartphones can be located through Wi-Fi and iBeacons attached to medical furniture with smartphones. This would provide an inaccurate but inexpensive, easy-to-deploy solution for locating furniture outside highly equipped rooms.

Based on this observation and the lack of existing solutions for combining different geolocation solutions, we have designed a new solution to solve this problem. Our solution tries to facilitate the integration of new location solutions. Its architecture is also designed to be distributed, support multiple users, and allow for multiple positioning levels.

The rest of this paper is organized as follows: Section II provides an overview of related research in geolocation systems that aim at combining several kinds of location technologies. Section III presents the proposed solution and its architecture. Section IV describes its technical implementation in detail. Finally, Section VI concludes the paper and provides directions for future work.

II. Related Work

By location systems, we actually mean very different technologies that aim at tracking people or goods, providing services intended for end users or infrastructure owners, for real-time service or for analytics [5][6]. In terms of hardware, the measurements are taken using users smartphones, or through antennas or sensors that are specifically deployed for this purpose, such as RFID readers and Wi-Fi routers. The underlying technologies are Global Positioning System (GPS) [7], Wi-Fi, Bluetooth Low-Energy (BLE), Ultra-Wide Band (UWB) [8], Light-Fidelity (Li-Fi) [9], video, Near Field Communication (NFC), Quick Response codes (QR codes), smartphone gyroscopes, compasses, etc.

The located devices are passive and active tags or smartphones, except for a few solutions such as video processing, which does not require any device to locate people and uses biometrics. Smartphones can act both as sensors and tags.

Two main categories of algorithms stand out. The first one is based on the estimation of the distance between the object to locate and several points whose positions are known (e.g. trilateration). Depending on the technology, the distance may be calculated from different metrics (Receive Signal Strength Indicator (RSSI), Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), etc.) and processed with basic algorithms such as trilateration or fingerprinting [2]. Because of the multipath propagation and non-line of sight, the results are quite poor. For these reasons, a lot of research is still underway to improve these techniques, such as the introduction of the Inertial Measurement Units and the use of Hiden Markov Model, grid-based filter, Kalman filters, particle filters, etc. [5].

The second category is based on the association of several physical measurements with a position. It requires a training/calibration phase (e.g. fingerprinting), which is quite heavy work since someone has go to the site and measure the different signals with different devices. Moreover, if the physical configuration of the site changes, this measurements are not relevant anymore. Research is being carried out on solutions for crowdsourcing these data in order to automate this task [10].

Hybrid technologies and signals of opportunities are promising solutions as they use every available source of information to improve accuracy and availability [11][12][13]. For example, they use smartphones 3-axis accelerometers and digital compasses to calculate the motion dynamics information and combine it with RSSI Wi-Fi positioning. These solutions also make use of signals from legacy systems such as radios, television sets, Wi-Fi and mobile phones to calculate positions within a few meters where GPS is not available.

All the aforementioned solutions are interesting technologies that aim at improving accuracy and robustness by focusing on the location of one object with one particular approach. Several of these proposals are mature enough and have already been deployed. Nevertheless, these solutions are developed in silos, and there is no seamless location service.

To overcome this technological limitation, several issues have to be addressed. One of them is the lack of a common solution for representing indoor/outdoor geographical data. To address this issue, the Open Geospatial Consortium (OGC) is working on IndoorGML [14] to model spatial information.

Another limitation is the lack of an architectural solution for combining different location technologies. The i-Locate project [15] focuses on providing a toolkit and a platform for using and managing several indoor/outdoor technologies such as Quuppa [16] and Galileo [17]. At the end of the project in 2016, the I-Locate *core localization service* should integrate those technologies and several services such as a location resolver, identity management or asset management. This open-source approach is very interesting, especially because it will be tested on site in several Italian hospitals. Nevertheless, based on the project specifications, the proposed solution is monolithic and heavy, as it is based on a 3-tier server architecture. This prevents the easy deployment of part of the data processing on mobile phones or other parts of the architecture, such as a gateway.

Moreover, even though the solution will make it possible to guide someone using different location technologies during a single journey thanks to its routing service, it will not provide any mechanism for combining technologies and making the most of all the available devices simultaneously.

Our work does not focus on improving existing geolocation techniques, but rather on efficient integration between different technologies. Compared to other projects with the same objectives we aim at providing a system with decentralized intelligence.

III. ARCHITECTURE

The whole architecture mainly relies on a *publish/subscribe broker*. The task of the broker is to receive messages and serve them to those who need them. The main idea is that antennas, locators and algorithms are *clients* of the *broker* (Figure 1). A client can subscribe to topics (channels) and publish messages on topics. When a client publishes a message on a topic, all the clients who have subscribed to that topic receive the message instantly. Sent messages can be marked as retained so they are kept on the *broker* and sent to *clients* when they subscribe to that *topic* later on.

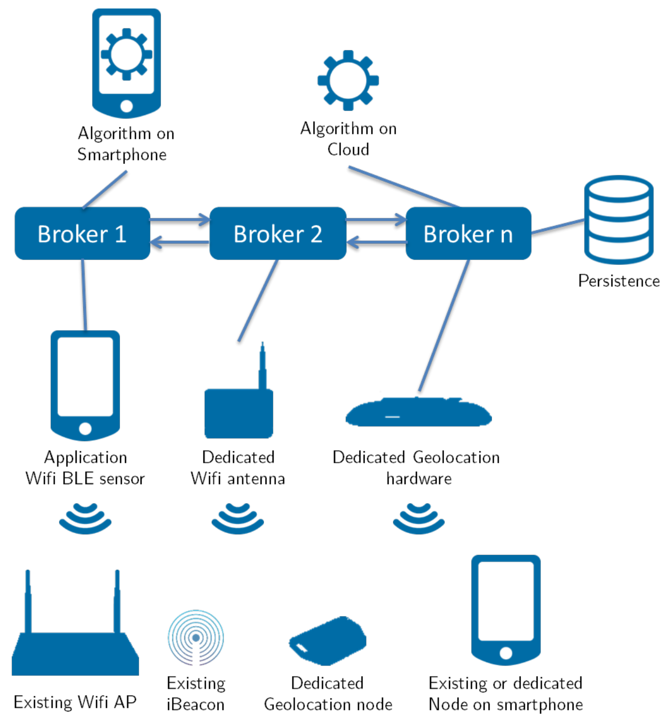

Figure 1. Architecture: all the blocks connected to the brokers are clients. The three clients under the brokers are called antennas.

With this architecture, we aim to decentralize everything as much as possible. Each sensor, algorithm or client can run on different devices as soon as they are connected to the broker. The broker can also be decentralized as will be explained later in Sections IV-A and VI.

A. Identifiers

To track nodes or users using multiple technologies, we need a way of doing so using these different technologies. For instance, a node may be identified by its Bluetooth and Wi-Fi Medium Access Control (MAC) addresses. To know that the node identified via Bluetooth and the node identified via Wi-Fi are the same, we need to declare that both identifiers are linked. To do so, a node or user must declare identifiers for all

 ISBN: 978-1-61208-421-3

the different technologies that it uses. To do so a message has to be sent on different topics. The message is always the same and is a dictionary of *Technology: Identifier* plus a *timestamp*:

```
{"BLE":"aa:bb:cc:dd:ee:ff",
"WIFI":"11:22:33:44:44:66",
"timestamp":1425384812592}
```

This message is sent to the following topics:

```
Identifier/BLE/aa:bb:cc:dd:ee:ff
Identifier/WIFI/11:22:33:44:55:66
```

These messages are retained on the *broker* so they only need to be sent when the device connects for the first time.

B. Measures

The measurements can be taken by any device that can sense its environment and communicate with the broker. Measurements can be Wi-Fi probe requests, iBeacons sensing, QR code or NFC tag reading, inertial measurements and so on. Measurements are sent in JSON format on a specific topic. For instance, the device with Wi-Fi Mac address `00:11:22:33:44:55` that receives a beacon packet from `aa:bb:cc:dd:ee:ff` with a signal strength of -59 will send the message:

```
{"timestamp":1422274308,"rssi":-58}
```

on the topic :

```
Measure/00:11:22:33:44:55/WIFI/aa:bb:cc:dd:ee:ff
```

C. Location requests

When a client wants to know its own location or that of another device (e.g. the location of an iBeacon), it has to request it. This way, if a device gets sensed by some sensors but nobody asks for its location, no algorithms run uselessly. Location algorithms are instantiated on demand. To request a location, a device needs to publish a retained message on the topic:

```
Location/Request/<requester>/<located>
```

where `<requester>` is the identifier of the client that requests the position, and `<located>` is the id of the device to locate. While the message exists on the broker, algorithms will try to compute the located position. When the locator does not need the `<located>` position anymore, it removes the message from the *broker* by sending a null message on the same topic to inform the algorithms that they can stop locating the device.

D. Referential

A referential is an abstract area associated with a position. A referential can contain an image of a map associated with an origin on the map and vectors. This way, there is no need to map an area to the World Geodesic System, and new areas can easily be added.

E. Algorithms

An algorithm is associated with an algorithm helper in charge of listening for location requests that involve algorithm. The algorithm helper can be in charge of several different algorithms. When the algorithm helper receives a location request, it verifies that the algorithms that it handles are capable of answering the request, and it eventually instantiates one or several algorithms to handle the location request.

When an algorithm is running, it listens for a certain type of measurements for the located device. The measurements are often associated with another device than the `<located>` one, which may be the locator or an antenna. To compute the located position, the algorithm may need the positions of locators or antennas. The algorithm will then request the location of that device. If the position is fixed, it may have been set as retained in the broker, so the answer is known instantly. Otherwise, another algorithm may answer the request. This mechanism makes multilayer positioning possible.

The position computed by the algorithm is sent on the topic: `Location/<located>/<algo_id>`.

F. Fusion

Several algorithms may compute the position of one device. They will post the position that they computed on the subtopic according to their names. In order for a final client to know its location, the results from the different algorithms have to be combined. A Fusion algorithm is in charge of combining the different positions computed by the different algorithms, and posting the merged position on the topic: `Location/<located>`. Like other algorithms, Fusion algorithms may be executed on a smartphone, server or any device that can connect to the broker.

G. History

In addition to the retained messages that the broker keeps, a history of measurements, locations and identifiers is kept. A specific client listens for all messages on the *broker* and stores them into a database. For now, this history is not used, but instead of waiting for measurements, *Algorithms* can request them in the history to speed up cold starts. The history may also be used to know the position of a device at a given time in the past.

IV. Implementation

To test our architecture, we have deployed several components: a Message Queue Telemetry Transport (MQTT) Broker, an Android mobile application, a modified wireless router and a cloud application server.

A. MQTT Broker

Our implementation relies on existing technologies and some that were specifically developed for the project. We use the open-source *Mosquitto* MQTT broker [18] for the *publish/subscribe* service. Several instances of the broker can be deployed on different servers to balance the load. Testing brokers capability is one of our ongoing tasks.

 ISBN: 978-1-61208-421-3

B. Android Mobile application

An Android mobile application has been written as a mobile client. Its purpose is to connect to the Broker and send iBeacon measurements. It also searches for Wi-Fi networks every 5 seconds in order to feed data into the wireless router.

C. Wireless router

Three TP-LINK MR3020 routers have been modified to listen for Wi-Fi Beacon frames and send them to a Broker. The *OpenWRT* [19] firmware has been installed on these routers and a custom application written in C connects to a Broker and is listening for 802.11 beacon frames. The application uses *libmosquitto* [18] and *libpcap* [20].

D. Cloud server application

Some algorithms and a fusion algorithm have been implemented on the server side. They are implemented in Java and connect to the MQTT broker. The algorithms implemented are trilateration over Wi-Fi, trilateration over Bluetooth, and a simple fusion algorithm that computes the barycenters of given positions.

E. In-browser viewer

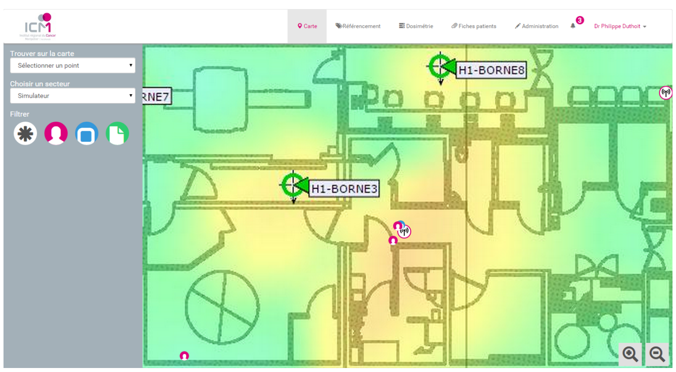

Figure 2. Web based viewer

The *mosquitto* MQTT broker can handle *websockets* connections; therefore, a web browser can be used as viewer. A web page has been developed that connects to the broker (Figure 2). The user selects an object to locate. Then the viewer displays the referential associated with the position and draws the object on it. A referential can also be selected so it is displayed, as well as all the objects located on this referential.

V. Results

The working architecture have been compared with the location service of an Android smartphone using Google Maps Android Application [21] and Quuppa [16] tags and antennas.

The test scenario is to start from a parking lot outside a building, enter the building and then enter a room equipped with Quuppa antennas. The building is also equipped with three Wireless routers described in Section IV-C. The test is made by a person holding an Android smartphone and a Quuppa tag. The smartphone have our application running in background and the Google Maps application in foreground. High precision location and Wifi is activated on the device.

On the Google Maps application the localization is accurate with an error of 5 meters when outside. As soon as the smartphone enter the building, the indicated position is off by up to 25 meters and the device is always located outside the building. As the device move in the building the indicated position barely move while staying outside the building, increasing the error of positioning.

On the Quuppa positioning system, the position reported by the system is not available until the tag is in the room equipped by the Quuppa antennas. In that room the error made on positioning the tag is less than 0.20 m.

Finally, using our architecture to locate the smartphone, we measure similar precision than using the smartphone location service outdoor while increasing it indoor. Indeed when the smartphone is in the range of the wireless routers the system computes an estimation of the position with an error of less than 10 m.

These early results show that the platform is able provide a continuous location service while using several technologies indoor and outdoor. We show that we can achieve continuity of positioning using a unique system while keeping a precision in the same order of existing systems used separately. For now Quuppa has not been integrated in the architecture but it is planed as a future work. It would lead to a positioning system combining strength of all positioning system we have at our disposal while providing ability to switch from one system to another seamlessly.

VI. Conclusion and Future Works

Future work will focus on features, security and privacy, third-party integration and performance. Planned features are: more compact messages, the ability for referentials to be positioned in other referentials, and running a MQTT broker on mobile phones. With a MQTT broker on mobile phones, if an algorithm is also running on the mobile phone, there will be no need to send messages to a broker located on a remote server. Actually, the brokers will communicate with each other and only transfer the messages that need to be transmitted.

Security and privacy will also be addressed using MQTT brokers capabilities. The objective is to fully isolate data so unauthorized users cannot see other users data and locations.

Finally, we plan to integrate the Quuppa [16] and Ubudu [22] geolocation technologies as black-box algorithms.

Through this article, we have proved the concept of a decentralized geolocation architecture that support multiple technologies and layers. This work will lay the foundations for future experiments with multiple technologies and devices, and will be a business enabler for our customers.

 ISBN: 978-1-61208-421-3

REFERENCES

[1] M. Kouhne and J. Sieck, "Location-based services with iBeacon technology," in Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on, Nov 2014, pp. 315–321.

[2] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012, pp. 269–280.

[3] E. A. Fry and L. A. Lenert, "Mascal: Rfid tracking of patients, staff and equipment to enhance hospital response to mass casualty events," in AMIA Annual Symposium Proceedings, vol. 2005. American Medical Informatics Association, 2005, p. 261.

[4] "EKAHAU," (Accessed: 2015.07.07). [Online]. Available: http://www.polestar.eu/en/use-cases/shopping-malls-large-retailers.html

[5] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," IEEE Communications Surveys Tutorials, vol. 11, no. 1, 2009, pp. 13–32.

[6] Z. Farid, R. Nordin, and M. Ismail, "Recent Advances in Wireless Indoor Localization Techniques and System," Journal of Computer Networks and Communications, vol. 2013, Sep. 2013, p. e185138. [Online]. Available: http://www.hindawi.com/journals/jcnc/2013/185138/abs/

[7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, Global Positioning System: Theory and Practice. Springer Science & Business Media, Apr. 2013.

[8] C. E. Baumann, L. Carin, and A. P. Stone, Ultra-Wideband, Short-Pulse Electromagnetics 3. Springer Science & Business Media, Apr. 2013.

[9] A. Lavanya, S. Umar, K. A. Kumar, and A. Kiranmai, "Light Fidelity: The New Wireless Communication Technology," Journal of Mobile Computing, Communications & Mobile Networks, vol. 1, no. 1, May 2014, pp. 20–23.

[10] H. Shin, Y. Chon, Y. Kim, and H. Cha, "A Participatory Service Platform for Indoor Location-Based Services," IEEE Pervasive Computing, vol. 14, no. 1, Jan. 2015, pp. 62–69.

[11] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, "A Hybrid Smartphone Indoor Positioning Solution for Mobile LBS," Sensors, vol. 12, no. 12, Dec. 2012, pp. 17 208–17 233. [Online]. Available: http://www.mdpi.com/1424-8220/12/12/17208

[12] W.-L. Li, R. Iltis, and M. Win, "A smartphone localization algorithm using RSSI and inertial sensor measurement fusion," in 2013 IEEE Global Communications Conference (GLOBECOM), Dec. 2013, pp. 3335–3340.

[13] C. E. Galván-Tejada, J. C. Carrasco-Jiménez, and R. F. Brena, "Bluetooth-WiFi based Combined Positioning Algorithm, Implementation and Experimental Evaluation," Procedia Technology, vol. 7, 2013, pp. 37–45. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017313000066

[14] Y.-J. Kim, H.-Y. Kang, and J. Lee, "Development of Indoor Spatial Data Model Using CityGML ADE," ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2, Aug. 2013, pp. 41–45.

[15] "i-locate," (Accessed: 2015.05.04). [Online]. Available: http://www.i-locate.eu

[16] "Quuppa - Do More With Location," 2015, (Accessed: 2015.05.04). [Online]. Available: http://quuppa.com

[17] "GALILEO - CNES," (Accessed: 2015.05.04). [Online]. Available: http://www.cnes.fr/web/CNES-en/1507-galileo.php

[18] "Mosquitto," 2015, (Accessed: 2015.05.04). [Online]. Available: http://mosquitto.org

[19] "OpenWrt," 2015, (Accessed: 2015.05.04). [Online]. Available: https://openwrt.org

[20] "Tcpdump/Libpcap public repository," (Accessed: 2015.05.04). [Online]. Available: http://www.tcpdump.org

[21] "Google Maps Android Application," (Accessed: 2015.07.07). [Online]. Available: http://www.google.com/maps/about/

[22] "Ubudu | Mobile Proximity Solutions for Businesses," (Accessed: 2015.05.04). [Online]. Available: http://www.ubudu.com/