Physical Web for Smart Campus Management

Giorgio Delzanno, Giovanna Guerrini, Maurizio Leotta and Marina Ribaudo
Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Italy

Keywords: Internet of Things, Physical Web, Beacon Technology, Indoor Location, Database Security.

Abstract: Physical Web enables smartphone users to interact with physical objects and locations through the use of beacon technology. Beacons are small devices placed on physical objects or at specific places that can be detected by users' smartphones when within a range of up to some tens of meters. In this way, users can receive notifications on their handset or associate their presence with a specific place, enabling indoor localization. In this paper, we present the design and the prototype development of a platform for Smart Campus management based on the Physical Web metaphor. This beacon-enabled platform provides services for the registration and analysis of student attendance and for the scheduling of lectures, classrooms allocation, and event notifications (e.g., notify students when teachers are in their office). The software prototype has been implemented using state-of-the-practice technologies such as Node.js, Android, and MySQL and has been preliminary tested in real setting in the context of the Computer Science Bachelor degree at the University of Genova obtaining encouraging results.

1 INTRODUCTION

The Google's Physical Web project[1] – center stage at the Google IO developer conference in 2016 – was conceived to enable smartphone users to interact with physical objects and locations through the use of the beacon technology. Beacons (Statler, 2016)[2] are low cost radio transmitters that typically transmit a unique ID on a regular interval, e.g., 100-1000ms, in a range of approximatively 30 meters. Bluetooth-enabled devices can detect a beacon and receive its corresponding identifier, following the so-called lighthouse metaphor. Smartphone applications can use such ID to signal their physical presence in the beacon vicinity to a remote server and the limited transmission range of these transmitters provides precise users localization.

The typical application domain of Physical Web is that of proximity marketing (Jeon et al., 2018). In this context beacons are located nearby specific products and smartphone applications, enabled to detect beacons, redirect the user towards Web sites with details on the products, brand, coupons, special offers, etc. Beacons have been applied in other domains like indoor localization (Huh and Seo, 2017; Zhu et al., 2012; Kaulich et al., 2017; Mackey and Spachos, 2017; He et al., 2017), crowdsensing in public transportation (Kang, 2017; Cianciulli et al., 2017), tourism (Sato et al., 2017), usage of public spaces (Ng et al., 2017; Purta and Striegel, 2017), support for elder people (Kashimoto et al., 2017), and so on. One of the main issues precluding a wider adoption of the Physical Web was the need to install native apps on each user's smartphone but, more recently, this limitation has been circumvented thanks to client APIs and by using browsers as delivery channel and show up notification, e.g. in the Android Notifications Manager[3].

Our application of the Physical Web eco-system is aimed at providing an *automated support* for the management of typical *Academic Campus services*. As an example, lecture attendance is often a relevant factor influencing the performance of academic students (especially bachelor). Intermediate tests are often used as a way to encourage students to actively attend lectures and take exams. For instance, in the Computer Science bachelor of the University of Genova, students are encouraged to sustain intermediate

[1]https://google.github.io/physical-web/

[2]Google Physical Web architecture has been supported with physical devices such as Estimote beacons and client API's.

[3]https://developer.android.com/reference/android/app/NotificationManager

Delzanno, G., Guerrini, G., Leotta, M. and Ribaudo, M.
Physical Web for Smart Campus Management.
DOI: 10.5220/0006959102770284
In *Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018)*, pages 277-284
ISBN: 978-989-758-324-7

tests during the courses. The goal here is to allow students to split an entire exam into smaller parts thanks to intermediate tests accessible only to students who attend at least 75% of the lectures. Of course, intermediate tests are not mandatory and those students that, for any reason, cannot attend the lectures can take the entire exam at the end of the semester.

In this paper, we present DidUp a software platform providing (physical) Web services and mobile applications for the registration and analysis of lecture/meeting attendance, for the scheduling of lectures, for classrooms allocation, and for events notification. The *main novelty* of our Physical Web application comes from the use of beacons. Indeed, differently from scenarios like proximity marketing oriented to a single-user experience, in the DidUp ecosystem beacons are used to detect, in almost real-time, the presence of a possible large number of students in a lecture room. When attendance is mandatory for taking exams such a system can improve the lecture workflow by removing the need of collecting signatures, writing codes/passwords on the blackboard for students detections, etc. Furthermore, the same architecture can be applied in every scenario that requires a verified list of attendees, for example in the case of meetings in which it is mandatory to generate on-the-fly reports.

A prototype of the DidUp system has been developed using the Node.js[4] server-side technology and Android for dedicated mobile app interfaces. The internal structure of Node.js allows, as required in DidUp, to handle a large number of connections in a short period of time (Düüna, 2016). Furthermore, Android OS provides natively support for beacons, for secure network connectivity, and for protecting device resources. In our system, private data are exposed to the server only after obtaining permissions from the users. The system has been tested in real setting in the context in two courses of the Computer Science Bachelor degree with more than 120 students overall. In this paper, we present the DidUp system requirements, design principles, implementation choices, and a preliminary user experience evaluation of the DidUp system.

Organization

The paper is organized as follows. In Section 2 and 3 we discuss the system requirements of the DidUp platform and its infrastructure, respectively. In Section 4 we briefly present the data model which constitutes the back-end of the components of the platform architecture, described in Section 5. In Section 6 we report the related work, while in Section 7 we discuss some results of a preliminary evaluation and address future research directions.

[4]https://nodejs.org/

2 SYSTEM AND INFRASTRUCTURE REQUIREMENTS

The DidUp platform is based on a combination of hardware and software solutions that must be deployed in physical spaces like lecture and meeting rooms in an academic campus. The infrastructure and system requirements can be summarized as follows.

1. The system should provide *precise indoor localization* of users via their smartphones. The required precision must be in the order of a few meters (i.e., to cover a lecture room).
2. *Registration* of student attendance must be done in *real-time*. The DidUp server must be reachable from every lecture room.
3. *Classrooms* must be viewed as *geofences*. Only students inside a classroom should be able to register their attendance at the lecture taking place in there.
4. The DidUp server must be available during office hours.
5. The app and Web user *interfaces must be accessible to non expert users* and must provide an online help menu.
6. The system should be *resistant to server failures*, e.g., by using multiple server instances on different machines.
7. Logged *data must be stored persistently* in a data storage system. Data storage must be accessible to the DidUp application and administrators, only.
8. *Users must be informed* that the server makes use of localization data.
9. *Users must give their permission* for releasing data stored in the device (IMEI).
10. *Data* stored in the DidUp server *should not be released to third parties*.
11. Students (and staff members) must be *uniquely identified via official credentials* such as the matriculation (staff) number assigned by the central administration.
12. Students and staff devices must be *identified uniquely using IMEIs*. Every student (staff member) must associate a unique device (IMEI) to the corresponding personal identifier.

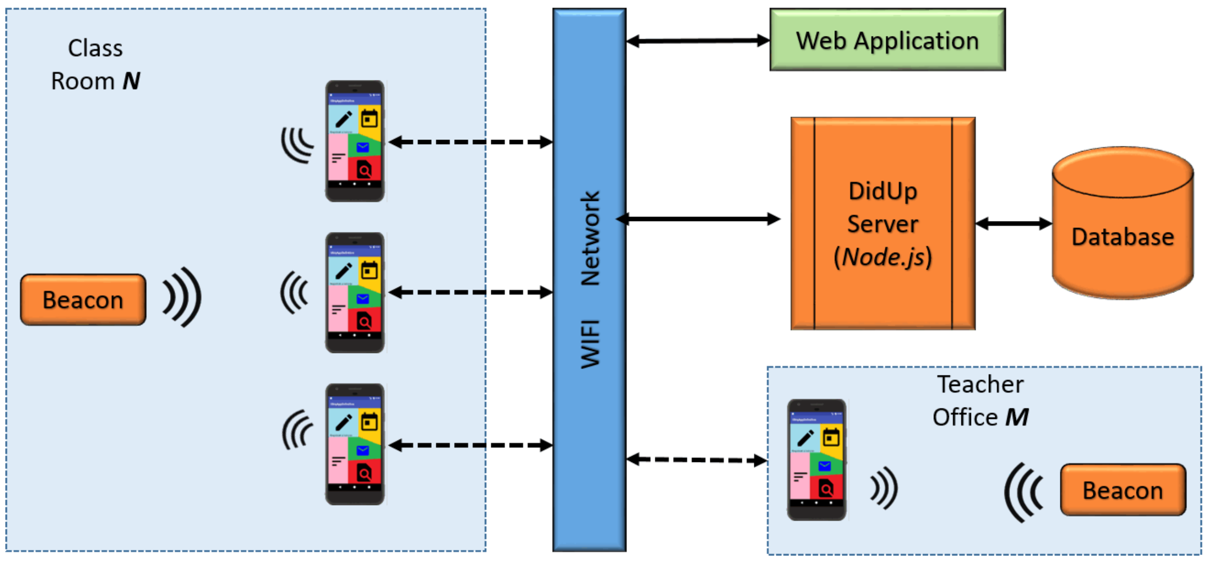

Figure 1: DidUp System Architecture.

13. *Data* exchanged with the users must be *encrypted and sent on secure channels*.
14. To *limit the budget*, hardware and software infrastructures must be composed by *low cost devices*, standard networking services provided in academic environments (e.g. server on virtual machines), open-source software and development frameworks for server-side and mobile applications.

To meet all the above requirements, we need to satisfy constraints at the physical (infrastructure and hardware), communication (networking), and software level (platform), as discussed in the following sections.

3 PHYSICAL AND HARDWARE ARCHITECTURE

The DidUp Physical Web system is based on the combination of beacons and wi-fi network technology. Our operative scenario is a typical example of an Academic Campus (see Figure 1) and we assume that each classroom is equipped with at least one beacon configured with a unique identifier consisting of three subfields called UUID, MajorID, and MinorID.

In our experimental setup we adopted Estimote beacons[5]. For lecture rooms for at most 180 students, as illustrated in Figure 2, a fix beacon turned out to be sufficient to detect all enabled smartphones. As an alternative to the use of hardware devices, it would also be possible to use smartphone apps that simulate beacons and that could be activated by the instructor at the beginning of the lecture. We also assume that teacher offices are equipped with beacons to inform students of their presence in the building and their office-hour updates. Finally, we assume that each room provides wi-fi access to students, e.g., via Eduroam[6] access points. A (Web) server must be installed either on an external cloud provider or on a local server. In both cases it must be reachable from the local network, i.e., the firewall configuration must provide access to the Web APIs needed by the DidUp smartphone and Web applications.

4 CONCEPTUAL MODEL OF THE DATA

In this section we describe the data model adopted in the system. We first illustrate some key ideas with the help of an example involving, for simplicity, only lessons attendance of students.

Assume that Alice Smith is enrolled in the first year of a Computer Science Bachelor degree. Alice has student ID 3471890 and the association between her identifier and the IMEI of her smartphone (and of other two students) is shown in Table 1.

The semester lecture schedule has two slots per day, namely 9-11 and 11-13 AM. We assume that the study plan of the first semester is given as shown in Table 2.

[5]http://www.estimote.com/Beacons

[6]https://www.eduroam.org/

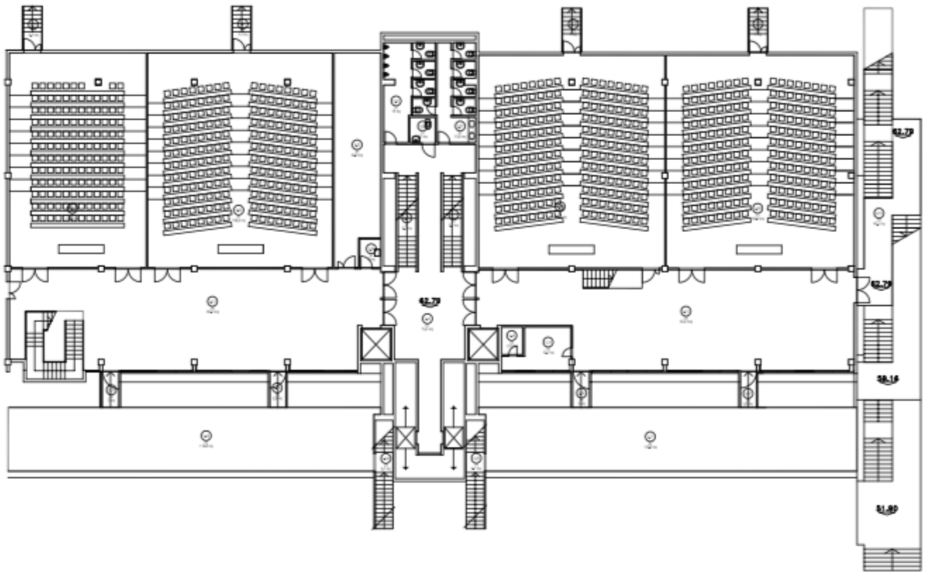

Figure 2: Lecture rooms planimetry (beacons were placed near the teacher's desks).

Table 1: Association between student ID and IMEI.

Matricula	IMEI
3471890	980000832471652
3471891	990000551621881
3471895	990000144425624

Table 2: Semester Lecture Plan.

	9-11	11-13
Mon	CS1 (room 1)	CS2 (room 1)
Tue	CS3 (room 1)	Lab1 (room 2)
Wed	CS1 (room 1)	Lab1 (room 2)
Thu	CS3 (room 1)	Lab2 (room 2)
Fri	CS2 (room 1)	Lab2 (room 2)

Table 3: Association between rooms and beacons.

Physical space	beaconID
room 1	101
room 2	102

The association between beacons identifiers and rooms is shown in Table 3. We now come to the requirements needed by the registration protocol. Attendance registration for Alice Smith is enabled twice every day and synchronized in accordance to the data in Table 2 and Table 3.

Timing is based on the server time in order to avoid manipulation of timestamps sent with user requests. As an example, in the 9-11 Monday slot the app installed on Alice smartphone scans the BLE network for beacon signal 101. If detected, it opens a connection with the DidUp server and, via the app user interface (a button), it provides the user a bridge to register attendance at the current CS1 lecture. In the 11-13 slot the same app starts searching for signal 102. Registration is disabled for the rest of the day and enabled again on Tuesday in the 9-11 slot for signal 101, this time associated to a lecture of course CS3, and so on.

Based on the above considerations, the data model of DidUp consists of the conceptual schema in Figure 3 (note that for the sake of readability, we show a simplified version of the ER schema including, for each table, only the most relevant information and focusing on the part of DidUp monitoring the students lecture attendance).

The `USERS` table contains data of students (and staff members) and associations with passwords (stored in hash form) and IMEIs. The `COURSES` table contains code, title, and year. `CREDENTIALS` associates login and password for accessing course data. `LECTURES` specifies records for each lecture (date, start, end). `ATTENDANCES` contains logs of individual lecture attendance (date and hours). `LECTURE_ROOMS` reports the name and size of each room available for the lecture and it size. `BEACONS` lists all the available beacons and associates each of them with a room.

5 SOFTWARE PLATFORM ARCHITECTURE

The DidUp software platform consists of an app with different views depending from the current user (e.g.,

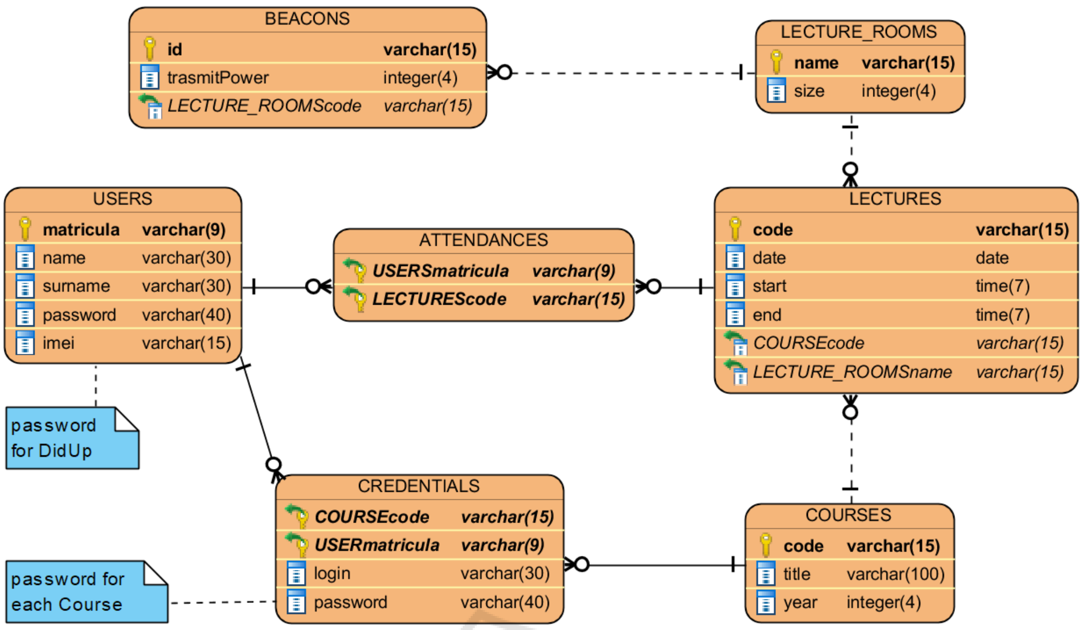

Figure 3: Conceptual Model of the Data (simplified fragment: teachers related part is not reported).

students, teacher, administrator), a Web application, and a persistency server as described in the next sections.

DidUp and Persistency Server

The DidUp server provides secure APIs (secure TCP) for accepting user requests and for sending notifications. Persistent data are stored in a MySQL server database accessible only from the server. For the model of persistent data see the fragment shown in Figure 3.

Web Application

The Web application allows staff members to create and modify user profiles, visualize historical data and statistics for both classrooms and individual students. Although responsive, this functionality must be viewed as an access point designed for a traditional (non mobile) browser.

Middleware

The middleware underlying the DidUp platform has been implemented combining different technologies. The server has been implemented using the Node.js IoT framework, an efficient server-side development framework based on JavaScript and on the npm[7] ecosystem. Node.js provides very efficient packages for handling secure TCP connections, Web servers, and applications. Node.js server-side libraries are optimized for network intensive applications even when executed on single host or cluster.

Indeed, Node.js allows to handle a large number of connections in a short period of time reducing potential risks of denial of service (this is very useful in our context since DidUp has to handle hundreds of requests in a few minutes). This property is due to the internal structure of the Node.js event-driven engine. Requests are not handled using multiple threads as in the Apache server model since the Node.js engine is based on an event loop that executes callbacks sequentially. Callbacks are picked from multiple priorities FIFO queues. Furthermore, thread pool implemented using the C++ libuv[8] concurrency library supports the execution of asynchronous callbacks. Differently from server architectures based on multiple threads, in Node.js the response to connection requests requires few system resources since they basically require the emission of events whose synchronous effect is that of enqueuing callback invocations in the I/O queue. This choice mitigates the risk of classical denial of service attacks based on a high number of simultaneous requests that could congest the server by exhausting system resources (Düüna, 2016) (e.g. creation of new threads to scale up the server).

DidUp App

The DidUp prototypes of the mobile applications have been developed in the Android OS. Android OS

[7]https://www.npmjs.com/

[8]https://github.com/libuv/libuv

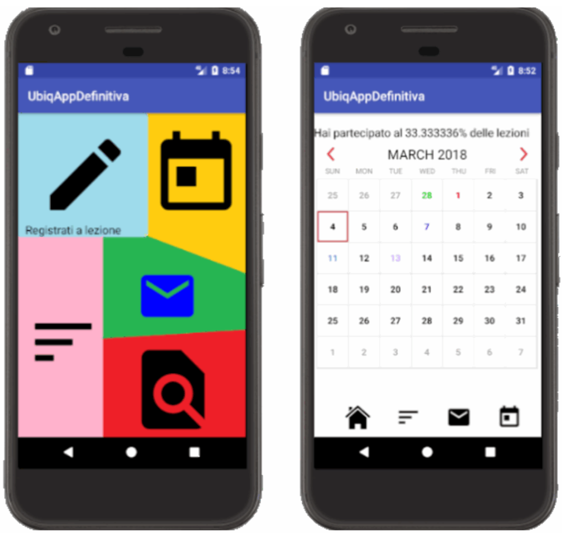

Figure 4: Student App: Mockup.

provides native support for beacons (e.g. using OS notification management or beacon SDKs like Estimote SDK). It also provides secure network connections and access control policy management that protects private data. Private data are exposed to the server only after obtaining permissions from the owner. The authentication mechanism built on top of associations between user and device identifier relies on secure network connections (secure TCP sockets) between the smartphone application and the DidUp server. A smartphone app, configured to detect beacons using libraries like the Estimote SDK[9], is provided in two different versions: students and teachers.

The *student app* provides sign-up, sign-in, and sign-out and a wide range of functionalities. Figure 4 illustrates mock-ups of the app: home activity (attendance is indicated via the pen icon), and the calendar activity (with different colors to indicate lectures and other events). The sign-up service associates the smartphone IMEI to the unique student enrollment number and to his contact details. After the first user registration, sign-in is automatically enabled whenever the app is activated by the user. Indeed, the server can retrieve the IMEI of the smartphone from the initial connection request issued by the app. Thus, the silent authentication stage is based on the association between the user identifier and the registered IMEI. Upon authentication, students can register their attendance to a given lecture in a given time-slot. The lectures schedule of each student is indeed synchronized, server-side, with the lectures and rooms allocations plan, for each enrollment year. Thanks to this synchronization, attendance is enabled only in specific time-slots and in physical spaces. In addition the app provides user interfaces for visualizing the historical data stored in the DidUp persistence server (profile, attendance log) and interfaces to visualize, via a calendar widget, lecture plans (according to the corresponding study plan), seminars, and other events registered by staff members.

Using protocols similar to those adopted for the student view, the *teacher app* provides a user interface for visualizing statistics on students attendance, lectures and events calendar. The aggregated number of presences provides indications for each course trend: it is indeed possible to understand if the number of students attending at the beginning of the course decreases significantly or remains stable during the semester. It is also possible to check if there are days or time-slots with a significant decrease in attendance and consequently try to implement strategies to avoid such drop out. The app also provides a control widget for the notification of presence in office-hours. Staff members can create, modify, and delete events that will be notified to users and visualized in their calendar view. Furthermore, they can access the room allocation service that is moderated by a dedicated administrator.

6 RELATED WORK

In this section we discuss other examples of beacon-enabled applications and compare them with our project proposal. In general bluetooth-based beacons make mobile devices aware of the surrounding environment (Gast, 2014; Statler, 2016; Jeon et al., 2018).

[9]https://estimote.com/

This technology enables a wide spectrum of applications such as cell-based localization enabled by intelligent placement or proximity-aware systems for the interaction with nearby objects. As an example, (Ito et al., 2015) presents a tour and navigation system based on proximity detection. The system provides the tourists time table of nearby bus stop and distance to nearby subway stations. (Ng et al., 2017) describes an interactive system for art galleries, which outperformed the conventional QR code's engagement conversion rate and time. Estimote has implemented a BLE-beacon based system in a museum to provide detailed information about an artwork to nearby users (Anderson, 2017). The system employs a pull mechanism, where the information is provided on request. The work in (Kang, 2017) is based instead on an infrastructure based on around 1000 beacon nodes across Hong Kong for push promotion and location advertising. Apple has implemented proximity based services such as AirDrop to allow iOS devices to connect to other devices in their vicinity. (Thomson, 2014) describes applications of iBeacons on a car for automatic transaction at toll booths, parking meters, gas station and more. BLE beacons have been mainly used to detect fine-grained location and movement to better identify the activity of the users with help of gesture detection technology of smart wearable devices. Knowing the user's micro-location helps to narrow down the list of possible gestures/actions users may take. Other applications to indoor localization can be found in (Huh and Seo, 2017; Zhu et al., 2012; Kaulich et al., 2017; Mackey and Spachos, 2017; He et al., 2017). (Kashimoto et al., 2017) presents a system that collects data of elder people. The system is based on wearable BLE beacon tags equipped with accelerometer. BLE beacon signals scanned by pre-deployed fixed scanners helps to identify micro-location of the user. Data coming from the accelerometer helps to identify the type of activities of the subjects under monitoring. Beacons are often used in combination with mobile crowd sensing technology. The main goal of these works is to replace expensive sensors on vehicles and street infrastructures with mobile crowd-sourcing that still enable safe driving experiences and awareness of car accidents and traffic information. (Cianciulli et al., 2017) described a distributed infrastructure for measuring traffic congestion, road conditions, parking availability, outages of public works and for real-time transit tracking. Examples of mobile crowd sensing applied to transportation can be found e.g. in (Chen et al., 2012; Sato et al., 2017). Beacons as technology support for improving the usage of physical spaces is discussed in (Purta and Striegel, 2017). Although the system architecture layer of our systems has common features with the above mentioned beacon-enabled solutions, we believe that our work has at least two main novelties: a new application domain and a novel combination of beacon and mobile device technology via Node.js.

7 CONCLUSIONS AND FUTURE WORK

The DidUp system is currently in alpha testing in some of the undergraduate courses of the Bachelor in Computer Science at the University of Genova. In particular, it has been adopted in the Computer Architecture and Database courses (respectively first and second year of the bachelor). Subscription was on voluntary basis since, as an alternative modality, students could still certify their presence to lectures by signing attendance sheets. As a result of this preliminary evaluation the service has been subscribed by 25% of the students. This is also due to the fact that part of the students refused to install the smartphone application mainly for privacy issues. We plan to improve the system, e.g., by providing a cross-platform Web-app and integrating other functionalities for improving the user experience and incentivate the adoption of the Physical Web app for attendance registration. We plan to extensively test the system in the academic year 2018/19 and to improve its quality by employing end-to-end automated testing solutions (Leotta et al., 2018b) and runtime verification techniques (Leotta et al., 2018a). Moreover, we plan to further extend our prototype to include also iOS devices in order to cover, virtually, the 100% of the students (note that no changes are required to the Node.js server). We also plan to improve data exploitation and integration with student/course data provided by our University. The resulting IoT framework is currently under evaluation as a possible physical infrastructure for access control based on a new specification language for location- and time-based RBAC policies (Delzanno and Guerrini, 2018) that can be automatically compiled into a set of permission rules that can be checked dynamically with an extension of the DidUp system.

There are many other interesting directions related to emerging beacon technologies like battery-free beacons (Wiliot, 2018), video estimote (Estimote, 2018), etc. In particular battery-free beacons could open new opportunities related to physical identification and authentication of users (e.g. attach beacons to badges, etc). Interactions with video estimote is also very promising since it can transfer personalization through user profiling, a typical feature of Web Applications, to the Physical Web. For example, a student wearing a passive beacon could receive personalized multimedia notifications in proximity of an information screen.

ACKNOWLEDGMENTS

The authors would like to thank Claudio Orlando for the prototype implementation of DidUp and Steve Statler for interesting discussion on beacons technology.

REFERENCES

Anderson, J. (2017). The icon of modern art puts estimote beacons on display. Available: https://blog.estimote.com/post/157200820650/the-icon-of-modern-art-puts-estimote-beacons-on.

Chen, X., Santos-Neto, E., and Ripeanu, M. (2012). Crowdsourcing for on-street smart parking. In *Proceedings of the 2nd ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet 2012, pages 1–8, New York, NY, USA. ACM.

Cianciulli, D., Canfora, G., and Zimeo, E. (2017). Beacon-based context-aware architecture for crowd sensing public transportation scheduling and user habits. *Procedia Computer Science*, 109:1110 – 1115.

Delzanno, G. and Guerrini, G. (2018). An IoT framework for context-aware role based access control. In *Proc. of the 26th Italian Symp. on Advanced Database Systems*, SEBD 2018.

Düüna, K. (2016). *Secure Your Node.js Web Application: Keep Attackers Out and Users Happy*. The Pragmatic Bookshelf.

Estimote (2018). https://estimote.com.

Gast, M. (2014). *Building applications with IBeacon: proximity and location services with bluetooth low energy*. O'Reilly.

He, W., Ho, P. H., and Tapolcai, J. (2017). Beacon deployment for unambiguous positioning. *IEEE Internet of Things Journal*, 4(5):1370–1379.

Huh, J.-H. and Seo, K. (2017). An indoor location-based control system using bluetooth beacons for IoT systems. *Sensors*, 17(12):2917.

Ito, A., Hatano, H., Fujii, M., Sato, M., Watanabe, Y., Hiramatsu, Y., Sato, F., and Sasaki, A. (2015). A trial of navigation system using ble beacon for sightseeing in traditional area of Nikko. In *Proceedings of IEEE International Conference on Vehicular Electronics and Safety*, ICVES 2015, pages 170–175.

Jeon, K. E., She, J., Soonsawad, P., and Ng, P. C. (2018). BLE beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2):811–828.

Kang, S. C. (2017). Apple daily launches in-app, beacon-based targeting. Available: http://www.campaignasia.com/article/apple-daily-launches-in-app-beacon-based-targeting/441080.

Kashimoto, Y., Morita, T., Fujimoto, M., Arakawa, Y., Suwa, H., and Yasumoto, K. (2017). Sensing activities and locations of senior citizens toward automatic daycare report generation. In *Proceedings of 31st IEEE International Conference on Advanced Information Networking and Applications*, AINA 2017, pages 174–181.

Kaulich, T., Heine, T., and Kirsch, A. (2017). Indoor localisation with beacons for a user-friendly mobile tour guide. *KI - Künstliche Intelligenz - German Journal on Artificial Intelligence*, 31(3):239–248.

Leotta, M., Ancona, D., Franceschini, L., Olianas, D., Ribaudo, M., and Ricca, F. (2018a). Towards a runtime verification approach for Internet of Things systems. In *Proceedings of 2nd International Workshop on Engineering the Web of Things (EnWoT 2018)*, LNCS. Springer.

Leotta, M., Clerissi, D., Olianas, D., Ricca, F., Ancona, D., Delzanno, G., Franceschini, L., and Ribaudo, M. (2018b). An acceptance testing approach for Internet of Things systems. *IET Software*.

Mackey, A. and Spachos, P. (2017). Performance evaluation of beacons for indoor localization in smart buildings. In *Proceedings of the 5th IEEE Global Conference on Signal and Information Processing*, GlobalSIP 2017, pages 823–827.

Ng, P. C., She, J., and Park, S. (2017). Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries. In *Proceedings of IEEE International Conference on Multimedia and Expo*, ICME 2017, pages 1069–1074.

Purta, R. and Striegel, A. (2017). Estimating dining hall usage using bluetooth low energy beacons. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp 2017, pages 518–523.

Sato, G., Hirakawa, G., and Shibata, Y. (2017). Push typed tourist information system based on beacon and augumented reality technologies. In *Proceedings of the 31st IEEE International Conference on Advanced Information Networking and Applications*, AINA 2017, pages 298–303.

Statler, S. (2016). *Beacon Technologies: The Hitchhiker's Guide to the Beacosystem*. Apress.

Thomson, D. (2014). ibeacon auto: Your car is a beacon. Available: http://beekn.net/2014/02/ibeacon-auto/.

Wiliot (2018). https://www.wiliot.com.

Zhu, J., Zeng, K., Kim, K. H., and Mohapatra, P. (2012). Improving crowd-sourced Wi-Fi localization systems using Bluetooth beacons. In *Proc. of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 290–298.