uBeacon: Configuration based Beacon Tracking

Prateek Jain*, Sanket Subhash Khanwalkar*, Rohit Malhotra*, Ajey Dheenrajappa*, Gaurav Gupta†, Alfred Kobsa*†

*Department of Informatics, University of California, Irvine
†Department of Computer Science, University of California, Irvine

***Abstract*—Beacons are small Bluetooth low energy (BLE) signal emitting devices that can transmit data about their location, temperature, motion etc. Various startups have commercialized Beacon based applications, such as for tracking valuable items, micro-location targeting of people, micro payments etc. However, these applications are not customizable beyond the use cases they were designed to serve. We feel that there is a need to build software that would allow users to configure the use of beacon data subject to the requirements of an application context, and to reuse the same set of beacons for some other purpose by just modifying the configuration. In this paper we describe uBeacon, a cloud based service with a web interface that allows users to define configurations that include constraints between beacon signal readers and beacons linked to objects or people. These constraints take the form of rules, such as "Object A should always be within 1m of object/reader B." uBeacon stores these configurations in a database, continuously monitors beacon data from readers, and notifies users via mobile push notifications whenever a rule of a configuration has been violated. We report the results of a pilot study with a set of participants and discuss the several possible applications of uBeacon to highlight its usefulness.**

Keywords—Ubiquitous Computing; Beacon; Location; Context Aware; Cloud based

I. INTRODUCTION

Location aware computing is the ability of a computing device to be cognizant of its absolute or relative location in space. While being useful on its own, location can also be employed to infer the activity a user is performing, social context and mode of transportation [1]. The global location based services market is expected to grow from USD 11.36 billion in 2015 to USD 54.95 billion in 2020 at compounded annual growth rate of 37.1% [2].

One of the approaches to determine the location of an object is to focus on its proximity to a known reference [1]. This is where beacon technology comes into play. A beacon is a Bluetooth low energy (BLE) transmitter at a known location which emits a continuous or periodic signal with limited information content [3]. The information content can include its identification, location relative to the reader, motion, temperature, etc. A reader can sense the presence of a beacon within a range of 10 meters or less. Signal strength is taken as a measure of proximity.

Beacons are easy to program. A number of established communication protocols have been developed to interact with them. The most notable ones are Apple's iBeacon for iOS and Google's Eddystone for Android. As [4] points out, over 570 million iOS and Android powered mobile phones are currently capable of receiving signals from Beacons. Multiple implementations of beacon technology are commercially available that serve a broad range of use cases in indoor location tracking, advertising and marketing.

Beacons are merely devices capable of transmitting information, but it is the context surrounding the use of that information that matters. Context comprises both social and physical phenomena concerning an activity [7]. For example, if we want to make sure that two prisoners don't come within 100m of each other due to their propensity to fight with each other, we must not only capture their absolute locations in space but also ensure the distance between them is less than 100m. This highlights the point made by Troshynski & Dourish [8] that space is not just absolute but is also constructed through the interaction between people navigating through it.

We believe that people should be able to reuse the same set of beacons for different contexts just by customizing the configuration between beacons (objects) and readers. They ought to be able to specify rules how readers should interpret data provided by beacons, be it location or temperature. Those rules reflect the context in which beacons are associated with objects that are tracked by readers. This is the motivation behind creating *uBeacon*, a novel cloud hosted service with a web based UI that lets users create rule based configurations between beacons and readers. A configuration is a set of user defined policies (rules) applicable to the objects to which beacons are attached and the reader(s) that track these objects. To give an example a policy could be "The wallet should always stay within the bedroom". Here the wallet is attached to a beacon and the beacon is being tracked by a reader installed inside the bedroom. The concerned user would be sent a notification whenever their wallet got closer to readers installed in other rooms than the one in the bedroom. Furthermore, users can change these configurations and their constituent policies anytime. So the same beacon can be associated with some other object or reader and become a part of a different configuration. Beacons thereby become decoupled from applications. This customizability is a key advantage of uBeacon over existing beacon related systems.

In the following sections we will refer to related work in the field of beacons, elaborate on the design and architecture of uBeacon, report the results of a pilot study to test uBeacon, and conclude with a discussion about the various real life applications of uBeacon and its future scope.

II. Related Work

In this section, we will refer to some of the research and industrial work done in the field of beacons. Given their small size and ability to be tracked by Bluetooth enabled readers, beacons are used for indoor location tracking/mapping, targeted advertisements, automatic mobile payments etc. In [9], Jones describes an indoor navigation system using low energy location beacons. Tile Co. has developed a tiny Bluetooth tracker to help people keep track of their valuable belongings [6]. Woolet Co. has developed a beacon enabled slim wallet that notifies its owner if he/she leaves it behind [10]. Biehl et al. [11] developed a trusted indoor location estimation system for mobile devices using stationary beacons. In the marketing domain, the retail chain Target has deployed beacons to deliver micro location based advertise-ments as push notifications [5]. In the payments domain, PayPal developed the PayPal Beacon [14], a BLE device that detects when shoppers running PayPal's mobile app enter a store. Points of sale are integrated with PayPal and allow "hands free" payments post check-in. Though structurally different from beacons, radio frequency identification (RFID) tags are also useful for tracking objects and people etc. Philipose et al. [12] explore installing RFID tags enhanced with sensors inside homes to track the daily activity levels of elderly people. The city of Aarhus in Denmark is currently undertaking a project to give bikers the ability to turn traffic signal on/off using bike installed RFID tags that can be detected by readers installed on signals [15]. Wayfindr is exploring the use of beacons to help visually impaired people navigate through indoors independently [16].

The difference between all these beacons based products and uBeacon is that the latter isn't about solving a specific location tracking or indoor mapping etc. problem, but about giving users the ability to configure how they want to use their beacons and providing an execution platform for their configurations.

III. Design & Implementation

In this section we will describe the design and technical vocabulary associated with uBeacon.

A. Technical Terms

- *Partial Object Context (POC)*: Anything attached to a beacon is termed an object. Partial object context is a triple of the form <Object, Reader, Property>, that describes the context (distance/location or temperature) of an object with respect to a reader that is tracking the beacon attached to that object. POCs describe a state. For example, if beacon B is attached to a wallet (object) and its relative location (property) is being tracked by reader R, then the partial object context for it is: <wallet, R, 1.0 >.
- *Policy*: A policy is a rule on an object's partial object context. It limits the range of valid POCs for an object. For example, if a user wants his/her wallet to remain within the room, then they can set a policy that the wallet must remain within (say) a 2 meter radius of the reader in that room. In this case, the set of valid POCs would be < wallet, R, 2.0 >. The cloud service sends a notification to the user anytime a policy is violated.
- *Configuration*: uBeacon allows users to define configurations, i.e. a set of policies applicable to one or more beacon tracked objects and readers. Configurations capture the real world motivation behind the use of beacons. For example if a user wants his/her belongings to stay within the bedroom, then they can define a configuration named "Bedroom" containing a set of (say) 5 policies that apply to 5 different beacon tracked objects with respect to the reader installed in that bedroom.
  <Wallet, R, 3.0 >, <Ring, R, 3.0 >
  <Bag, R, 3.0 >, <Laptop, R, 3.0 >
  <Mobile, R, 3.0 >
  The reader sends location data to the cloud hosted service, and the service checks if any object is out of place. It sends a notification to the user in case any object is farther from the reader than it should be. To give another example, users can create a configuration to track the temperature of a set of flammable objects (to which beacons have been attached) and name it "Anti Fire". A reader receives temperature data from all these beacons. Anytime the temperature of any of the objects belonging to this configuration exceeds the specified limit, the cloud service sends a push notification (as a fire alert) to the user's mobile phone.
- *Configuration Management*: Configuration management refers to creating, editing and deleting configurations of beacon attached objects and their associated readers. uBeacon currently provides a web based UI to create configurations and manage the policies underlying them.

B. Architecture

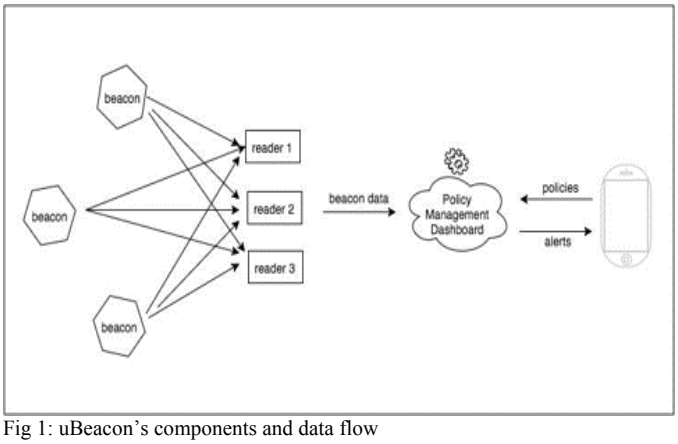

Fig 1: uBeacon's components and data flow

uBeacon's architecture consists of three main components:

- *Beacons*: As mentioned previously, beacons are simple Bluetooth low energy transmitters. Users attach beacons to objects for tracking their location,

temperature etc. The beacons transmit a beacon id, signal strength, and temperature to readers. We convert signal strength to distance using the Estimote SDK [13].

- *Readers*: Readers are devices capable of receiving and processing Bluetooth signals. We considered using Raspberry Pi and Arduino as readers for this research, but chose android powered smartphones instead because they are ubiquitous and can upload Bluetooth data to the cloud. The phones run a background task that receives Bluetooth data from the beacons and sends it to the cloud hosted service. If readers are kept stationary, they can be associated with the room they are in, in which case room id can be used instead of reader id.
- *Cloud Service*: The cloud based RESTful service receives beacon data from all the readers. It stores this data in a MySQL database and checks to see if it violates any configuration. If it does, then the service sends a push notification [see Fig. 3] onto the uBeacon mobile app that is installed on the user's phone.
- *Web Dashboard*: uBeacon's web based dashboard [Fig: 2] allows users to log in to the system, associate objects with beacons, create and manage configurations/policies etc.

C. Frameworks/APIs

The following frameworks have been used for implementing uBeacon's various components:

- *Android Software Development Kit (SDK)*: uBeacon's mobile based service runs on Android and is built using the Android SDK.
- *Estimote SDK*: Estimote SDK provides the API used by the mobile based service for interpreting beacon data.
- *Google Cloud Messaging (GCM)*: uBeacon's push notifications (alerts) are powered by GCM.
- *Volley*: uBeacon's mobile service uses Volley for sending asynchronous web requests containing beacon data to the cloud service.
- *Django REST framework*: uBeacon's cloud hosted service is implemented using the Django MVC framework and is hosted on Amazon EC2 public cloud.

IV. Evaluation

We conducted a small study consisting of 12 participants (7 male and 5 female, all aged 20-26 and students at UCI). We installed an Android phone running uBeacon's mobile service as a reader in a room of their choice. Then we asked the users to perform the following tasks in a stepwise manner:

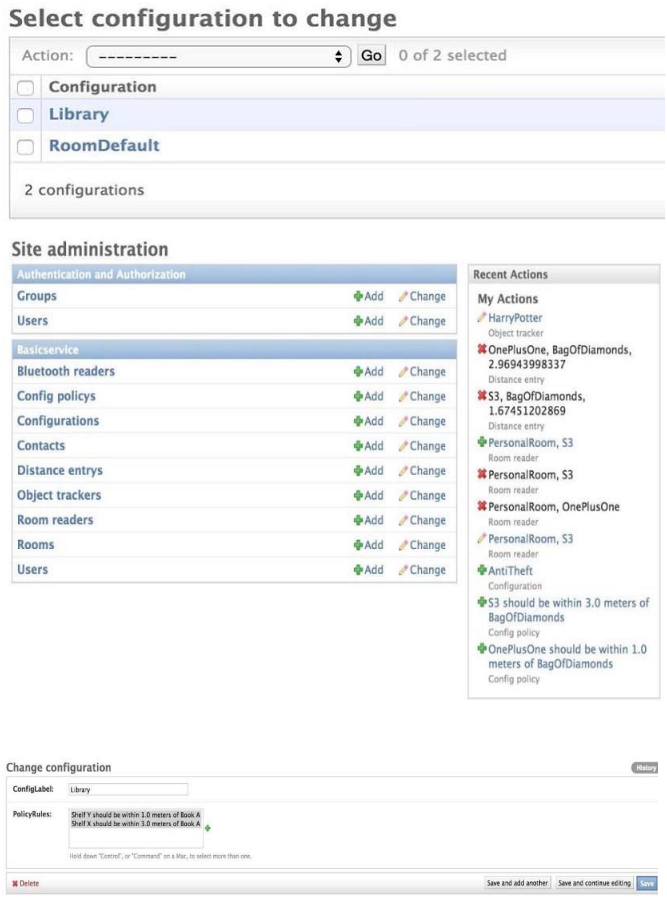

Fig 2: The Web Based Dashboard

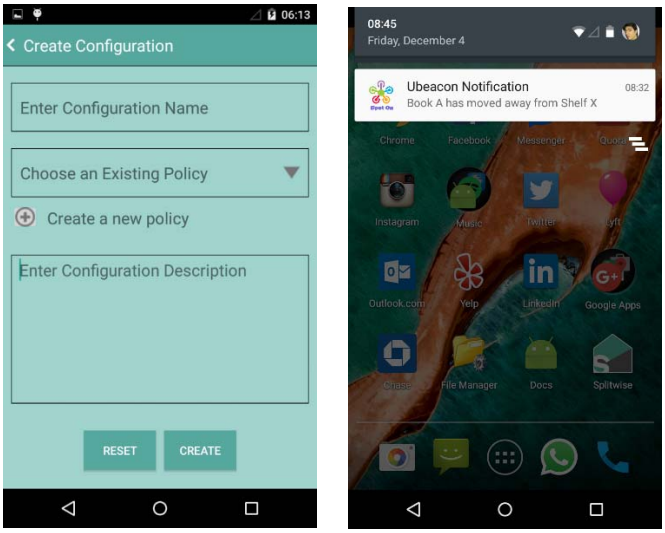

Fig 3: Mobile app showing the push notification & the UI for creating a configuration

1. Register an account in uBeacon using the web dashboard.
2. Attach an Estimote beacon to any valuable belonging.
3. Create a distance related *Anti-Theft* configuration between room installed reader and the beacon (that is attached to the object) using uBeacon's web dashboard.

4. Purposefully take their belonging outside the room.

It took about 15 minutes for each participant to complete the above mentioned steps. Upon successful completion, all participants received a push notification on their registered mobile phones alerting them that their belonging has been displaced. We then asked them several yes/no and subjective questions to judge their experience, concerns and willingness to try uBeacon as a product in the near future.

Question	**Y**	**N**
Are you willing to accept uBeacon as a product that tags and tracks indoor location of your belongings?	10	2
Would you use uBeacon (which has access to locations of your personal belongings) in shared spaces like apartments, offices, universities, etc.?	7	5
Would you recommend uBeacon to your friends?	10	2
In future, would you prefer uBeacon over prevailing indoor location tracking products like Tile?	8	4
Would you prefer if uBeacon were tightly integrated with beacon hardware? (like Woolet)	3	9

Based on the results, we feel that privacy of indoor location data is an area of concern for users. More work needs to be done to see how uBeacon can be deployed in shared spaces.

V. USE CASES AND FUTURE WORK

We developed a cloud based service that allows users to set up and manage configurations between beacons (objects) and readers. Configurations that can represent a variety of real life scenarios. Our system uBeacon can be used in any context that requires a set of objects (personal belongings or otherwise) to be in a particular place or arrangement with respect to others. One such place is a library. If every shelf has a reader and every book is tracked by a beacon, then librarians can use uBeacon to create a *Bookshelf* configuration that notifies the librarian whenever someone puts a book in the wrong place. Another place could be a chemical laboratory where it is necessary to keep substances such as sodium and water at a distance from each other. In this case, the containers can be labeled with beacons and a configuration can be set that alerts lab managers anytime they are mistakenly placed too close to each other. Inside the home users can label their valuable belongings with beacons and have a reader installed in their room. They can create *Anti-Theft* configurations using uBeacon that would notify them whenever a belonging left the room, based on increase in distance from the reader. Another interesting use case could be in residential parking lots. In this case, the residential services could give beacons to cars owners and place a reader at every parking spot and set up *Parking Lot* configurations using uBeacon that notifies car owners if someone else parks a car in their spot.

As future work we plan to incorporate code to triangulate an object's actual indoor coordinates (instead of just calculating relative distance) by taking data from at least 3 different readers. This would allow for representing more complex location based contexts as configurations. We are also working on developing a mobile app front end that would allow users to create configurations and manage them in addition to the web based dashboard. Based on privacy concerns raised during evaluation we would also like to test uBeacon in shared spaces (offices, café's etc.), and conduct a survey to gauge people's perceptions regarding sharing indoor location data with a remote app.

REFERENCES

[1] Jakob Bardram, Adrian Friday, Ubiquitous Computing Fundamentals, Chapter 7.

[2] Markets and Markets, "Location based services (LBS) and real-time location systems (RTLS) market by location (indoor & outdoor), technology, software, hardware, services, and application areas - global forecast to 2020".

[3] Introduction to beacons. [online]:http://www.slideshare.net/goldengekko/an-introduction-to-beacons

[4] The Beacons Report: Growth forecasts for the most important retail technology since the mobile credit card reader [online]: http://www.businessinsider.com/beacons-are-the-most-important-new-retail-tech-2014-7#ixzz39com8qiC

[5] Target launches beacon test in 50 stores, will expand nationwide later this year. [online]: http://techcrunch.com/2015/08/05/target-launches-beacon-test-in-50-stores-with-expanded-rollout-later-this-year/

[6] The tile app, lost and found stories. [online]: https://www.thetileapp.com/blog/lost-and-found-facts

[7] Paul Dourish, "What we talk about when we talk about context ", Personal and Ubiquitous Computing, volume 8 issue 1, February 2004, pages 19-30.

[8] Emily Troshynski, Charlotte Lee, and Paul Dourish, "Accountabilities of presence: reframing location-based systems", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 487-496.

[9] Edward Forster Jones Jr., "Indoor navigation with low energy location beacons" , Patent US 20140228044 A1, August 14, 2014.

[10] Woolet smart wallet. [online]: http://techcrunch.com/2015/03/13/the-woolet-is-a-wallet-that-yells-at-you-when-you-leave-it-behind/

[11] Jacob T. Biehl, Adam J. Lee, Gerry Filby, Matthew Cooper, "You're where? prove it!: towards trusted indoor location estimation of mobile devices" UbiComp '15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pages 909-919.

[12] M. Philipose, J. R. Smith, B. Jiang, A. V. Mamishev, S. Roy, and K. Sundara-Rajan. "Battery-free wireless identification and sensing". IEEE Pervasive Computing, 4(1):37-45, 2005

[13] Estimote SDK for Android. [online]: https://github.com/Estimote/Android-SDK

[14] PayPal launches beacon, 'Hands Free' in-store payments hardware. [online]: http://www.paymentssource.com/news/technology/paypal-launches-beacon-hands-free-in-store-payments-hardware-30153731.html

[15] These RFID tags allow Danish cyclists to turn traffic lights green. [online]: http://www.fastcoexist.com/3053969/these-rfid-tags-allow-danish-cyclists-to-turn-traffic-lights-green

[16] Wayfindr. [online]: http://techcrunch.com/2015/12/03/wayfindr-open-standard/

[17] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Dmitriy Portnov, Roy Want, "Bridging physical and virtual worlds with tagged documents, objects and locations", CHI '99 Extended Abstracts on Human Factors in Computing Systems