BiDaE Technology

BiDaE Positioning System (BPS) Platform
An IOT Fog for Fine-Scale Indoor Location Specific Services

February 2023

1. Overview

This note describes the structure, components and distinct features of the *BiDaE Positioning System, or BPS* for short. It is an open platform designed to be a scalable, easy to deploy, enhance and maintain infrastructure for both IPIN (Indoor Positioning and Indoor Navigation) and real-time indoor object tracking applications. Figure 1 shows its structure and the applications and services supported by it: The top left part of the figure depicts *Seeing-I-Go* mobile apps and *BOT* (*BiDaE Object Tracker*): They are BiDaE Technology's own products. Through the open BPS API's [1], the platform also can provide indoor position and real-time object location data to diverse applications and services developed by others. The right top part of the Figure 1 shows several current and future applications and systems as examples.

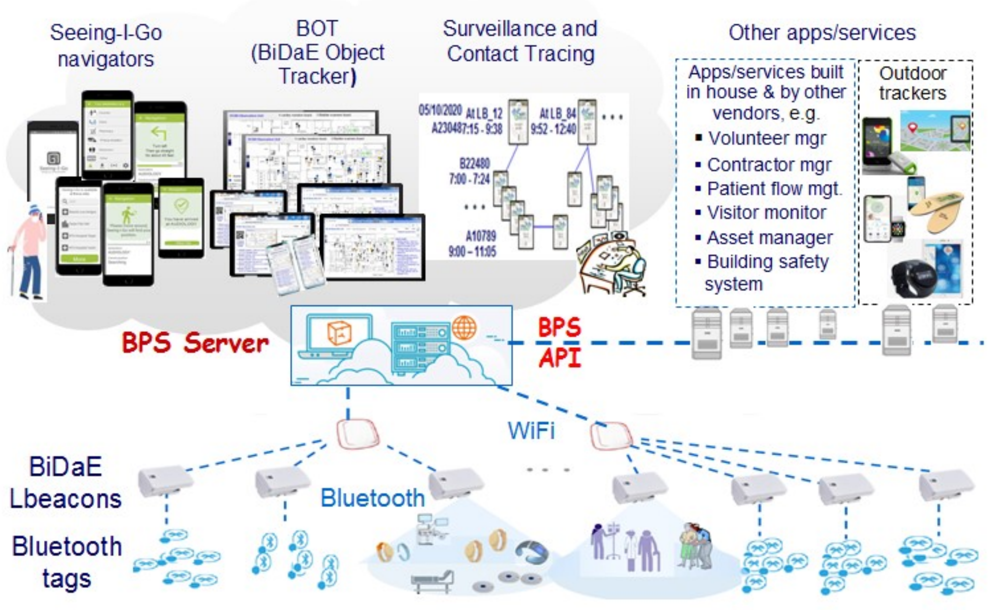

Figure 1 Structure of BPS platform/infrastructure and applications/services supported by it.

From Figure 1 one can see that *Lbeacons* (*Location beacons*) and *tags* are pervasive components. Both are Bluetooth low energy devices. We will describe them shortly. The platform is structured as an IoT (Internet of Things) fog. It remains responsive when overloaded in the presences of thousands of objects to be located and tracked in real-time. It degrades gracefully when network connection is disrupted and parts of it damaged. Built on a high availability architecture and structured to support high concurrency, the server is designed to be available 99.99 % of the time and can handle hundreds of concurrent searches via browser-based user interfaces. The table below lists these distinct features together with other figures of merit.

Table 1 Distinct features of BPS platform

- **Dual functionality** – Can serve as infrastructure/platform for both IPIN (Indoor navigation and indoor navigation) and real-time indoor object tracking APPs/services
- **Good performance**
  - Location accuracy – Can provide 3-5 m (bed/desk-level), 5-10 m (room-level), and 10- 20 m (zone-level) location accuracy
  - Response times – Can support 2 sec GUI response time and 10 sec delay in location updates
- **Scalability and robustness** – Location accuracy and interactive response time are affected negligibly by presence of thousands of objects and concurrent searches.
- **Configurability and enhancement** – Can be easily configured, deployed and enhanced to support different functions and achieve different accuracies.
- **Openness** – Open API's enable APPs and services developed by others to rely on BPS for indoor positioning and tracking support.
- **Maintainability** – Health conditions of Lbeacons and battery status of tags can be monitored centrally and Lbeacon health restored remotely.
- **High availability and resiliency** – Can provide 99.99% or better uptime and indoor positioning service degrades gracefully.

Following this introduction, the remainder of this notes describes the individual components of the platform and in passing, explains how the merits listed in Table 1 are achieved.

2. Tags and Lbeacons

Again the work done by BPS is carried out collaboratively by two types of components: They are tags and Lbeacons as illustrated by Figure 2.

Figure 2 Lbeacons and tags

(A) Simple and Sensor Tags

Tags are Bluetooth Low Energy (BLE) transmitters. Every tag has a universally unique identifier (UID). It broadcasts continuously advertizing packets containing its own UID and thus makes its own presence known to Bluetooth receivers near by. Real-time object location and tracking systems use of this feature to make objects to be located and tracked visible to the system. This is done by attaching to (or wearing by) every object to be visible to the system and storing in the server an association of the tag ID with the object ID/name. Thus, the location of each visible object is given by the location of the associated tag.

BOT (BiDaE Object Tracker) provides two types of tags: simple tags or sensor tags. Figure 2 shows some of their shapes and sizes. *Simple tags* do nothing but broadcasting their UID's. As the name indicates, each *sensor tag* contains sensors. In addition to its UID, each sensor tag also broadcasts readings of its sensor(s) as payload of the advertizing packets. Currently, BOT uses as sensor tags smart watches containing vital sign sensors: Each watch reads regularly once every 5 - 10 minutes vital signs of the person wearing it, including temperature, blood pressures, heart rate and blood oxygen level. BOT makes use of this capability to support remote vital sign monitor. Some sensor tags can provide acceleration readings. BOT uses them to monitor movements of persons wearing them. We can add other types of sensor tags (e.g., Bluetooth mobile air quality sensor) to provide additional remote sensing capabilities as needed.

(B) Lbeacons

Throughout the area where objects are to be located and tracked and navigation to be supported. Lbeacons are pervasively installed usually on ceilings. Figure 2 shows its appearance and how it looks when it is under ceilings.

Because of similarity between their names, one may think that Lbeacons are similar to iBeacons from Apple and Eddystone from Google. In fact, Lbeacon cannot be more different. Their differences include that Lbeacons are powered by AC or PoE (power over Ethernet) and broadcast their own 3-D coordinates via directional antenna. In contrast, iBeacons and other Bluetooth beacons are battery powered and broadcast their URL's or UIDs via omni-directional antenna. A more important difference is that iBeacons, Eddystones and so on are simple Bluetooth Low Energy (BLE) transmitters. In contrast, Lbeacons are not only BLE transceivers. Some of them also provide gateway function to link Lbeacons nearby to the WLAN.

Again, each Lbeacon stores and broadcasts its own 3D coordinates to mobile devices within a coverage area a few meters in diameter. Thus, it provides navigation app's on the devices with position data. Seeing-I-Go, the waypoint-based navigation application from BiDaE, uses as a support infrastructure Lbeacons deployed at locations where navigation directions are needed.

Each Lbeacon also scans its coverage area continuously and communicates via WiFi or Ethernet to the BPS server. Whenever a Lbeacon hears a tag, it sends to the server the UID of the tag, its own 3D coordinates, and the time interval during which the tag is heard. Based on the data from all Lbeacons installed in the area covered by the platform, the server determines at any time the location of every tag and the time interval in which the tag has been at the location.

Every Lbeacon also forwards to the server time stamped sensor readings from every sensor tag. In this way, Lbeacons together effectively function as a sensor data crowdsourcing system.

(C) Location Accuracy

BOT specifies the vertical location of every object by the floor level on which the object is. The vertical location of every object reported by the system is always accurate.

The system can be configured to achieve a wide range of horizontal location accuracy by making use of more or fewer Lbeacons. Oftentimes, BOT is required to bring the user to within sight of sought-after objects: To meet this requirement, Lbeacons are configured to achieve no more than 3-5 meters or 5-10 meters horizontal accuracy. This is often referred to as *bed/desk-level* or *room-level* accuracy, respectively. In hospitals and other healthcare facilities, this configuration is typically used for areas such as emergency department and patient wards.

A BOT may be configured to provide zone-level accuracy (i.e., a lower location accuracy of 10-20 meters or more) in all or some areas. This setting is usually used when the goal is to determine whether the sought-after object is in the building, or on which floor, in which patient ward or which part of a ward. Because fewer Lbeacons are needed to cover areas requiring zone level and floor-level accuracy, the cost of BOT is reduced accordingly.

(D) Lbeacons/Gateways as Edge Processors

In addition to being a transceivers, Lbeacons are embedded processors that carryout data processing at the edge of a IoT cloud: Each Lbeacons processes UID's extracted from advertizing packets of tags to eliminate duplicates, timestamps UIDs of tags to be sent to the server, generates its own health report and error messages, and so on. By doing this work, Lbeacons relieve the server from CPU intensive data processing tasks and reduce significantly the volume of data between Lbeacons and the server. This is a reason that the BPS platform is scalable.

As stated earlier, some Lbeacons function as gateways. They collect location data and health reports from Lbeacons nearby and forward the data via WiFi or Ethernet to the server. As part of gateway functions, these embedded processor prioritizes the transmissions of data from different Lbeacons according to the required response time, e.g., scheduling the transmission of data indicating alert conditions at the highest priority and transmissions of Lbeacon health reports at the lowest priority. As a consequence, BOT is able to deliver time critical alerts such as alerts from geo-fences within 1-2 seconds.

3. High Availability and Graceful Degradation

BPS platform is designed not only for scalability in ways described above, but also to be highly dependable. Dependability is achieved by building the BPS server and storage unit (hereafter referred to collectively as the server) on a high availability architecture on the one hand, by enabling the remainder of the system to degrade gracefully on the other hand.

(A) Graceful Degradation and Health Monitoring

Specifically, graceful degradation of BPS follows straightforwardly from the fact Lbeacons broadcast their own coordinates and collect UIDs of tags in their coverage areas essentially independently of one another. When a Lbeacon failures, the system can rely on Lbeacons in its vicinity to provide degraded service. For example, in an area where Lbeacons are configured to provide bed (or room) level accuracy, location accuracy may degrade to room (or zone) level when one or more Lbeacons failed. To provide added dependability, each Lbeacon automatically and continuously assess its own health during normal operation and send their health reports to the server periodically. Consequently, faulty and defective Lbeacons can be identified on a timely basis. Oftentimes their health can be restored remotely before failure occurs.

Similarly, battery status of every tag tracked by the platform is monitored continuously. BOT uses this capability to support low-battery alerts, letting designated user knows whenever any tag is low on battery and thus preventing failures to find objects due to dead tag batteries.

(2) High Availability (HA) Architecture

As one can see from Figure 1, BPS server would be a single point of failure that must be eliminated by providing redundancy for all but an acceptable length of downtime. For this reason, the server is built on a HA architecture. The design goal is to achieve equal or better availability that is typically required of hospital information systems (HIS), that is in the range of 99.95 to 99.99% availability. In other words , total downtime per year is shorter than 4h 22m 58s and 52m 35s, respectively.

Figure 3 shows the HA architecture for the server. Specifically, dual redundancy of the BPS server enables the system to tolerate single failure: It contains two nodes, one being active, while the other works in Hot-Standby mode. Each node has RAID-5 storage on which tried-and-true replication mechanisms, including PostgreSQL Replication Mechanism, can be used to duplicate the latest records. This configuration is easy to maintain and troubleshoot. Given the high reliability of modern server hardware and server and database software, this configuration should enable the platform to meet the 99.95 - 99.99% uptime requirement.

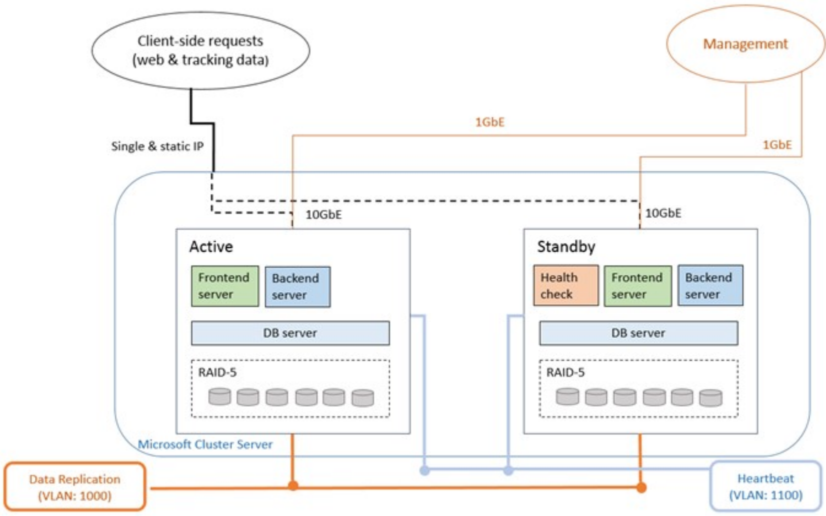

Figure 3 High Availability architecture for BPS server and storage unit

References

[1] BiDaE Object Tracker API SDK Document, January 2022, available by request