Feed Me

Satvik Dhandhania
Anand Sankar A

Recap

- Leveraging beacons placed in a stadium to streamline food delivery to audience members
- Strategically placed beacons detect mobile phones in its vicinity and allow people to place orders
- Customers are organized into zones based on their location
- Suggests the most profitable zone to the hawker on his phone to increase sales
- Customers can know when the food hawker is nearby

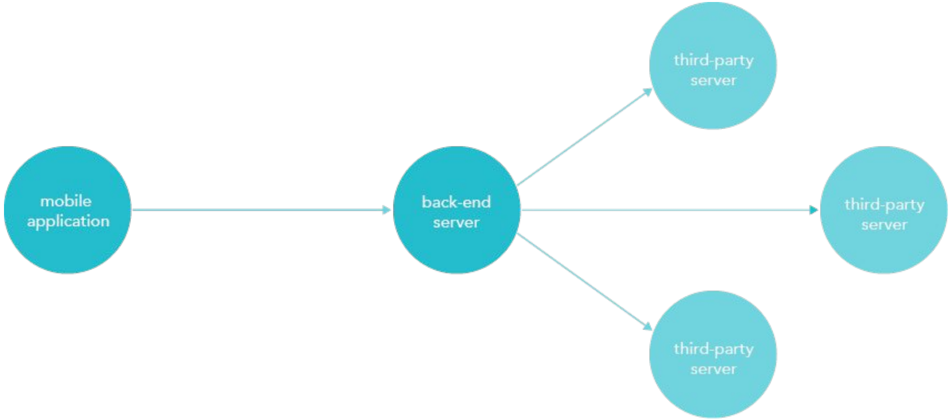
Gimbal Beacon - Series 10

- Bluetooth powered device that provides context about proximity
- Beacons broadcast a secure code to ensure authenticated user access
- Applications using Gimbal Proximity SDK can detect beacons in close proximity
- Configured to transmit at a rate that optimizes application’s reactiveness and battery life

Android Phone

- Bluetooth 4.0 enabled Android Device
- Internet Accessibility
- Gimbal Application with Permissions
- Android 6.0 or less

Third Party Server

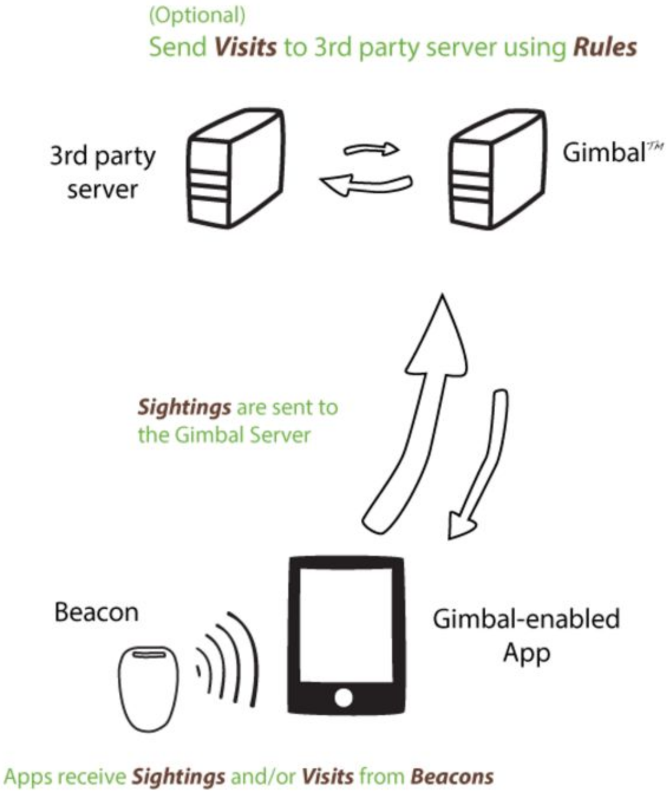


- Third Party Server holds states

- This server can be hosted on
  - Amazon Web Services
  - Google App Engine
  - Heroku

DEMO

Big Picture

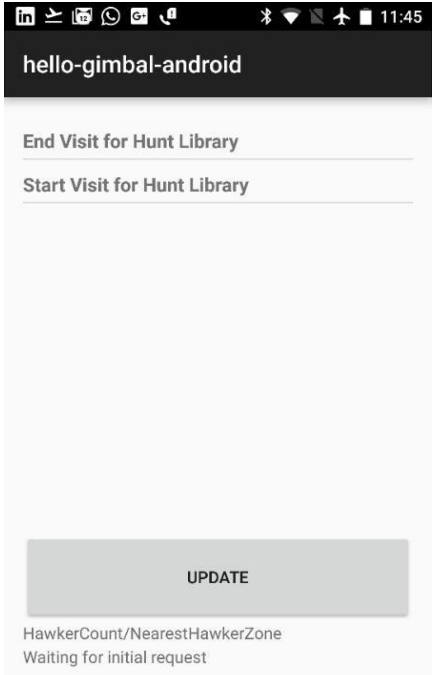

(Optional)
Send *Visits* to 3rd party server using *Rules*
3rd party server
Gimbal™
Sightings are sent to the Gimbal Server
Beacon
Gimbal-enabled App
Apps receive *Sightings* and/or *Visits* from *Beacons*

Sightings and Visits

- Sighting occurs when a proximity enabled application receives a BLE packet from a proximate beacon
- Visits can be used to characterize the amount of time spent around a beacon
- Characteristics of visit:
  - *Arrival RSSI:* The signal strength value that triggers an arrival event
  - *Departure RSSI:* The signal strength value at which the user is declared to be out of range of the beacon
  - *Departure Interval:* Determines how long the mobile device has to be out of range before a departure event is triggered

Assumptions in the Application

- A hawker or a consumer can only be at only one location at any particular.
- They should be out of the other zones they entered.



11:45

hello-gimbal-android

End Visit for Hunt Library

Start Visit for Hunt Library

HawkerCount/NearestHawkerZone
Waiting for initial request

Client Application

Challenges-1

- Different devices have different ranges and delays (RSSI)
- Setting Up server to communicate with the Device
- Making SDK to work with the Newer Devices (Android Vs Gimbal)
- Algorithm to work simultaneously with both the Audience and Hawkers.
- Maintaining the state on the server (Cross Zone State)

Challenges-2

- Handling cases where exits happen after a device has entered another zone.
- Maintaining a valid state of the server even with the inconsistencies of the beacon, while they continuously go in and out of the range
- Handling duplicate requests due to the above mentioned reason
- Tracking locations of all the hawkers and the clients
- Mapping nearby beacons to direct consumers towards beacon based on proximity if there are no hawkers servicing in his zone. (The customer knows where the nearest hawker is)

Challenges-3

- Invalidating requests that were inconsistent like queries due to interaction with multiple beacons which violates our base assumption.

- Keeping consistent state on the android device with interactions

- Running around to test the whole application and the use cases ;)

Use Cases

- Hawker and Consumer in the same zone

- Hawker and Consumer in Different Zones

- Hawker in a low profit Zone

- Consumer without any Hawkers at any locations nearby

Thank You!