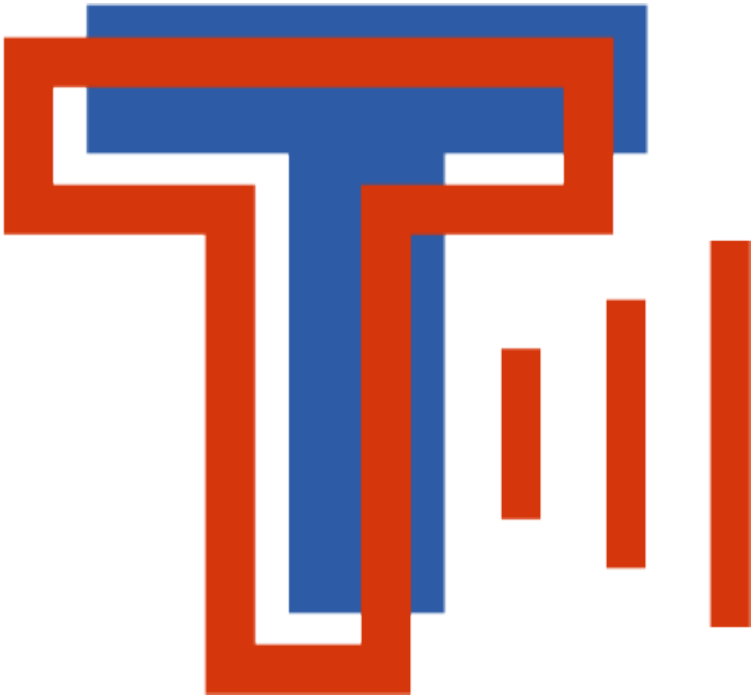
# Department of Computer Science & Engineering
# The University of Texas at Arlington

## Detailed Design Specification
## CSE 4317: Senior Design II
## Spring 2020



## Transact
## Translocate

KOPAWID SARAWICHITR
NOLAN BOWDEN
PRANAV BHANDARI
JIAN MA
AN NGUYEN

## Revision History

| Revision | Date | Author(s) | Description |
|---|---|---|---|
| 0.1 | 10.01.2019 | KS | document creation |
| 0.2 | 10.05.2019 | KS, NB, PB, JM, AN | complete draft |
| 0.3 | 10.12.2019 | KS, NB, PB, JM, AN | release candidate 1 |
| 1.0 | 05.15.2020 | NB | official release |

# CONTENTS

## List of Figures



## List of Tables

# 1 INTRODUCTION

Our product is a Location Based Messaging engine. The concept of this product is to provide users with messages based on the location they are currently at. This product will use a system of location gathering technologies, a mobile application, and a cloud infrastructure to accomplish this task. How our product will work is as such, the location information is gained by the use of a combination of BLE and GPS which then is received by a mobile application, the app will then send the location data along with the user identity information to then be processed in the cloud, the cloud will then publish the user information and the corresponding location, so subscribers will be able to send a message to the user.

## 2 System Overview

This project have two major components. There is a mobile application and a cloud infrastructure. Within the mobile app side, there are two app that are were design. App number one deal with the subscriber and app number two handle the location detection. On the cloud side, there is a Pub/Sub trigger system, a location configuration system, and an admin portal. With the combination of these systems, we are creating a location based messaging engine.

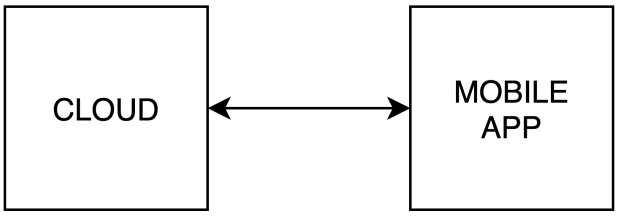



Figure 1: An overview of the software architectural design diagram

### 2.1 Cloud Layer Description

The Cloud layer will be developed to handle the location data received from the mobile devices, and then process and send that data to the event service. The data will flow within the cloud infrastructure through multiple subsystems. The server will be the first place the data will reach, here is where the data is received from the mobile device and sent to the Pub/Sub Trigger subsystem. The subscribers will receive the user identity and location from the pub/sub subsystem. The data will then move onto the Location Management Subsystem. The Location Management subsystem will determine if the user is in an existing location or not. It will check the database for validity. If all the data are valid they will receive a text massage. Admin Portal is an HTTP website which control all locations within the system. Admin can add, update, and delete locations and BLE beacons. All locations will be stored in Admin database. When a location is add, delete or update, admin database will also be updated. Then all admin interface will send a the update information to the location configuration.

### 2.2 Mobile Layer Description

The role of the mobile layer is to receive the signals from the location detecting technologies and send the data to the cloud, and for subscriber to register and sent messages to be add to the queue in the cloud. The Location Detection App will receive signal from the BLE Beacon, containing the BLE identification information's, and will get the GPS information's from the the mobile phone internal software. The data is then passed onto the Data Processing subsystem for packaging to be sent to the cloud. Before the packet is sent to the cloud, additional relevant device information is added. After processing stage is finish, the packet is sent to the Cloud Connection subsystem to be sent to the cloud. The Subscriber App will allow subscribers to register for an account to send message to a certain location. The Location detected layer is a key point for this project. The equipment for location detection are GPS and the BLE beacons. The equipment will detect the userâs phone and send that location information to the mobile phone application. The application will send data to the cloud server for next step data processing.

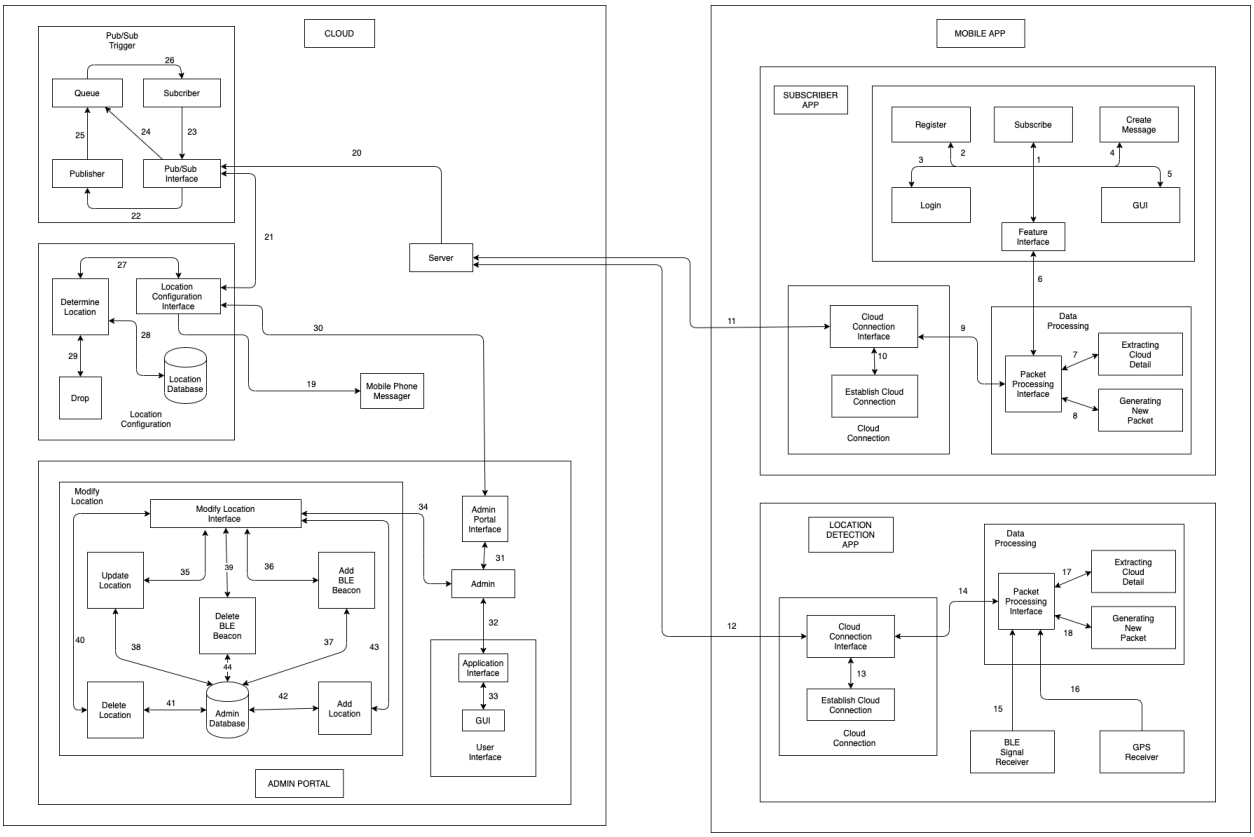



Figure 2: Data flow diagram

# 3 Cloud Layer Subsystems

The Cloud layer will be developed to handle the location data received from the mobile devices, and then process and send that data to the event service. The data will flow within the cloud infrastructure through four subsystems. First, the mobile service subsystem, where the data is received from the mobile device and sent to the location management subsystem. Then the Location Management subsystem will determine if the user is in an existing location stored by our system and if so send the data to be published by the pub/sub subsystem. The subscribers will then receive the user identity and location from the pub/sub subsystem and then the event service subsystem will send the necessary data to the user. Admin Portal is an HTTP website which control all locations within the system. Admin can add, update, and delete locations. All locations will be stored in Admin database. When a location is add, delete or update, admin database will also be updated. Then all admin interface will send a the update information to the location configuration.

## 3.1 Server

The server will be communicating between the cloud and the mobile app. Data coming from the app will contain multiple location information and the individual personal's information. The data will than be send to the relevant subsystem in the cloud. The data is received in the first server and processed in the other subsystems.

### 3.1.1 Assumptions

All the data coming into the subsystem is in the correct structure and is valid.

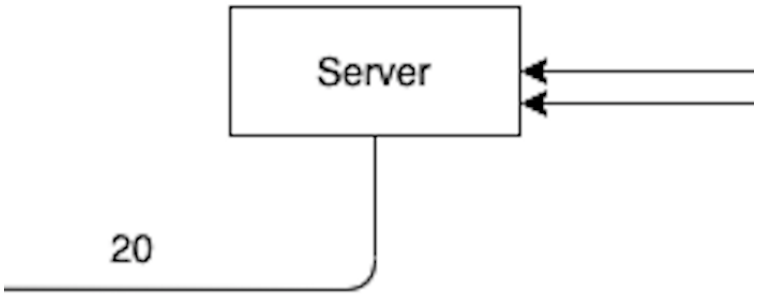



Figure 3: Server description diagram

### 3.1.2 Responsibilities

The responsibility of the sever is to host and communicate with the other subsystems within the cloud as well as receive data sent from the mobile application.

Table 2: Mobile Handle Service Subsystem

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #SV1 | Handle and host communication from the cloud and other platforms | Input 11<br>Input 12 | Output 20 |

### 3.1.3 Operating System

The OS of the VM running the server is Ubuntu 18.04

### 3.1.4 Software Dependencies

The nodejs library Express was used in this server.

### 3.1.5 Programming Languages

The server was written using nodejs.

### 3.1.6 Data Processing

The location information is received on the server through the use of HTTP POST requests, adn then the data is forwarded to a data storage structure powered by Azure Blobs.

## 3.2 Location Configuration Service Subsystem

Location Configuration Service Subsystem will configure/determine a user location based on the data it receives from the Pub/Sub Subsystem. The data received will be looked up in a database to determine if it is a valid location we are monitoring. if the location exists a message will be sent to the mobile device via text message. If we do not have a location that matches the data then the subsystem will drop the location performing no further computations.

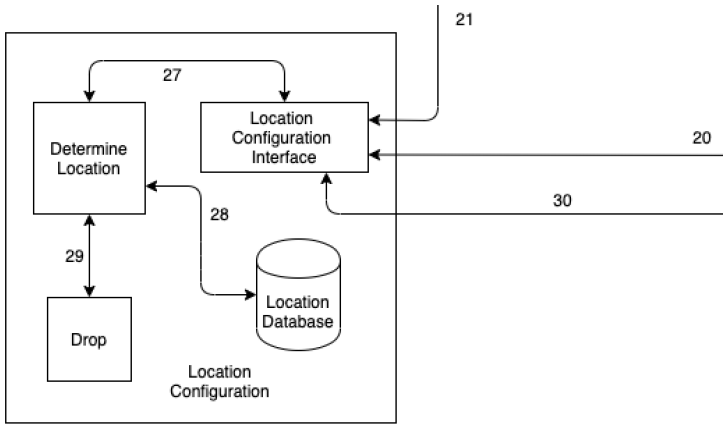



Figure 4: Location Configuration subsystem description diagram

### 3.2.1 Assumptions

All the data coming into the subsystem is in the correct structure and is valid.

### 3.2.2 Responsibilities

The responsibility of the Location Configuration Interface is to communicate with all of the other subsystems within the cloud. The responsibility of the Determining Location process is to query a database of locations known and determine if that location holds any significance. The Drop process is used if the location is not needed for use by the application, this process will cancel any further computation involving the received location data. The Location Database is responsible for containing all known locations that can be used by the engine, meaning that there is information that can be sent to a user as a message involving that location.

### 3.2.3 Software Dependencies

.NET CORE

### 3.2.4 Programming Languages

This subsystem was written using C-Sharp

### 3.2.5 Data Structures

This subsystem interacts with a database which stores the beacon information, GPS coordinates, and appropriate location description information.

### 3.2.6 Data Processing

This subsystem was implemented through Azure Functions. That data is pulled from a queue using Azure Blobs and then compared to locations stored in a database powered by cosmos-db. If the data matches a location then a text message is sent to the number corresponding to their user information stored in the database.

## 3.3 Pub/Sub Trigger Subsystem

The Pub/Sub Trigger system exists within the cloud. Data is first received in the server, and after the server collects the data it will publish that data into a queue created on Azure. This queue will contain all location information sent to the server from the mobile devices. The data will be consumed from

the queue by an Azure function designed to trigger anytime location data is published. After the data is consumed in the queue it is processed by the location configuration subsystem.

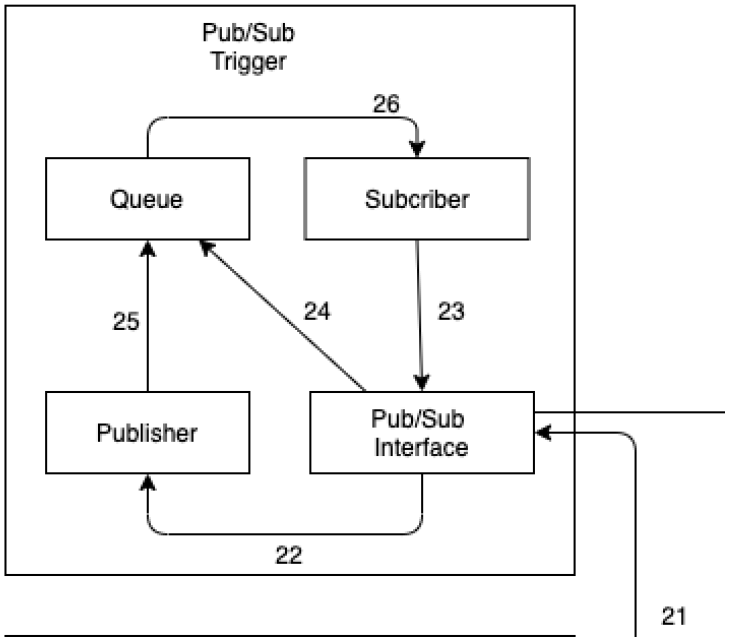



Figure 5: Pub/Sub subsystem description diagram

### 3.3.1 Assumptions

All the data coming into the subsystem is in the correct structure and is valid.

### 3.3.2 Responsibilities

The responsibility of the Pub/Sub Interface is to communicate with all of the other subsystems within the cloud. The responsibility of the publisher is to publish specific data to the cloud for a finite amount of time. The publisher will publish both the user information and their corresponding location. The responsibility of the queue is to store all of the data that is currently published. This data will persist within the queue for a specific amount of time, and if the subscriber is subscribed to a specific location corresponding to a user in the queue then it will consume that data to be used for a message. The responsibility of the subscriber is to consume user information held within the queue if they are in the specific location they subscriber has subscribed to.

### 3.3.3 Data Structures

This subsystem uses a FIFO queue created using Azure Blobs.

### 3.3.4 Data Processing

The queue in the pub/sub subsystem is a data strucure created using Azure Blobs, when location data is received in the server it is sent to this queue. When location data enters this queue an azure function is triggered to consume this data and use it to check the location and send a message.

## 3.4 Admin Portal Interface

Admin interface will handle all location configuration subsystem.

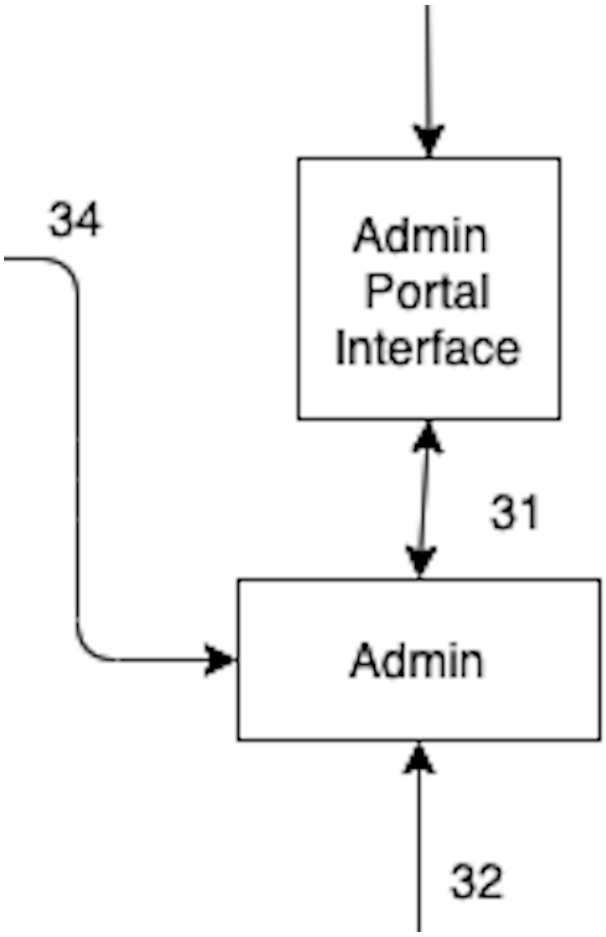



Figure 6: Admin portal diagram

### 3.4.1 Assumptions

Admin portal will only take input and produce output from and to location configuration.

### 3.4.2 Responsibilities

Admin portal interface is where all the communication happen within admin portal layer. Admin portal interface get notified from Admin on any update on location. Admin portal interface then send the update information to the location configuration.

## 3.5 Modify Location

### 3.5.1 Assumptions

N/A

### 3.5.2 Responsibilities

Admin will receive request to update, delete or add location and BLE beacons from the user interface. Modify Location interface then will forward this request to according service. These services will then update the database. The database contain all the information regarding BLE beacons and GPS for a certain location. Beacon have detail regarding their range and UID. GPS contain details about their latitude, attitudes, and longitude.

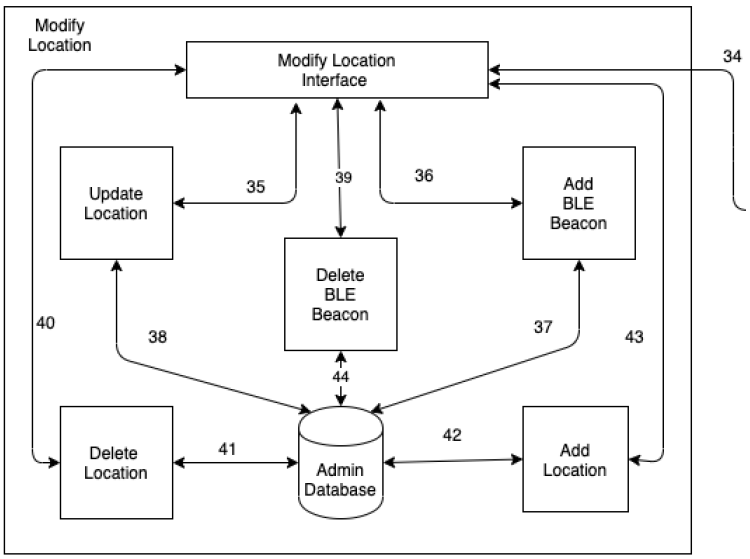



Figure 7: Event Grid
This is where location being modified.

### 3.5.3 Programming Languages

The languages used to implement the admin page are HTML, CSS as basic JavaScript, and nodejs

### 3.5.4 Data Processing

The admin page takes user input and stores that information in the database for locations.

## 3.6 User Interface

User Interface is where user make request through GUI

### 3.6.1 Assumptions

User input valid value. User know how to use GUI to input value.

### 3.6.2 Responsibilities

User interface include 2 part: GUI and application interface. GUI is used by direct user to make request to modify location. The request then sent Application interface. Application interface took the request and forward to admin.

# 4 Mobile Layer Subsystems

The role of the mobile layer is to receive the signals and data from the location detecting technologies and send the data to the cloud.The two IOS mobile app are the Location Detection App and the Subscriber App. The Location Detection App will receive signal from the BLE Beacon, containing the BLE identification information's, and will get the GPS information's from the the mobile phone internal software. The data is then passed onto the Data Processing subsystem for packaging to be sent to the cloud. Before the packet is sent to the cloud, additional relevant device information is added. After processing stage is finish, the packet is sent to the Cloud Connection subsystem to be sent to the cloud. The Subscriber App will allow subscribers to register for an account to send message to a certain location. The Location detected layer is a key point for this project. The equipment for location detection are GPS and the BLE beacons.

## 4.1 Mobile Layer Operating System

The mobile app is deployed on iOS

### 4.1.1 Mobile Layer Programming Languages

The mobile app was implemented in Swift.

## 4.2 Feature Interface Subsystem

The user interface subsystem handles the data received and also displays the data received from the cloud to the graphical user interface. It allows an individual to register to become a subscriber, login, subscribe to locations, and create message to sent out to publisher. After processing the user input, the updated data is then sent to the Cloud Connection subsystem.

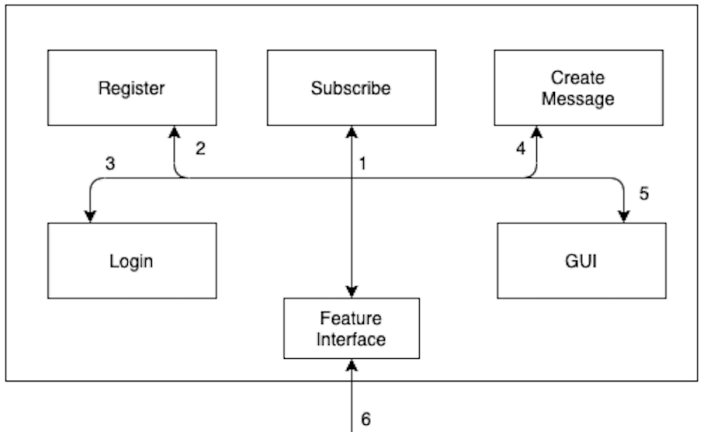



Figure 8: User Interface subsystem description diagram

### 4.2.1 Responsibilities

This subsystem is responsible for GUI and user actions.

### 4.2.2 Data Processing

This subsystem will display data received as notifications or representation in the mobile app user interface.

## 4.3 Data Processing Subsystem

This subsystem is identical in both the Subscriber and Location Detection App. This subsystem will process any data leaving and entering the mobile app. It will extract and generate new packet to be transmit.

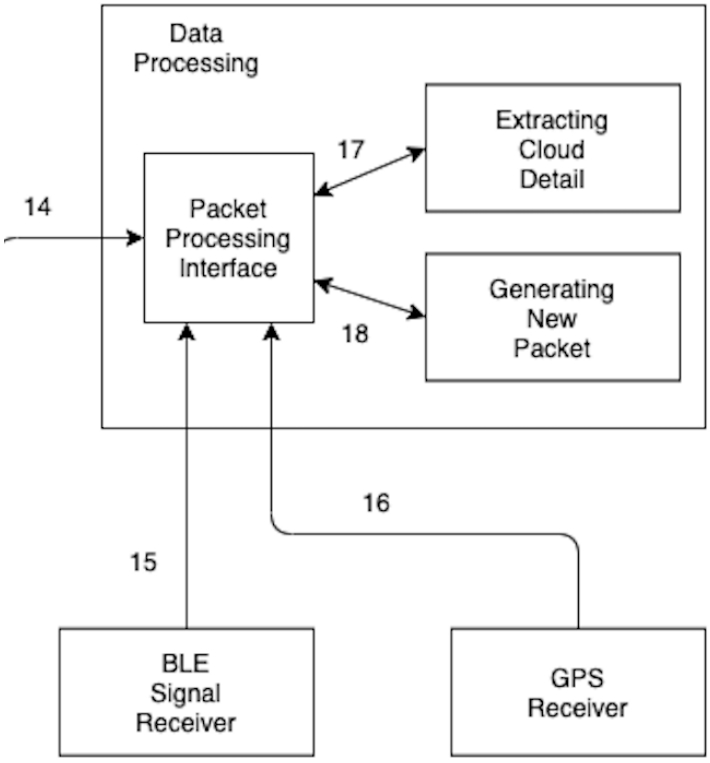



Figure 9: Data Processing subsystem description diagram

### 4.3.1 Responsibilities

This subsystem is responsible for processing the data received and generating new packet.

### 4.3.2 Data Processing

This subsystem takes data received and unpacks it to be used by the app, it also prepares data that needs to be sent to the server.

## 4.4 Cloud Connection Subsystem

The Cloud Connection subsystem is responsible for sending the data to the cloud. First, a secure connection to the cloud is established. Then the data is sent to the cloud for further processing. This subsystem is identical in both the Subscriber and Location Detection App.

### 4.4.1 Assumptions

The cloud details are valid

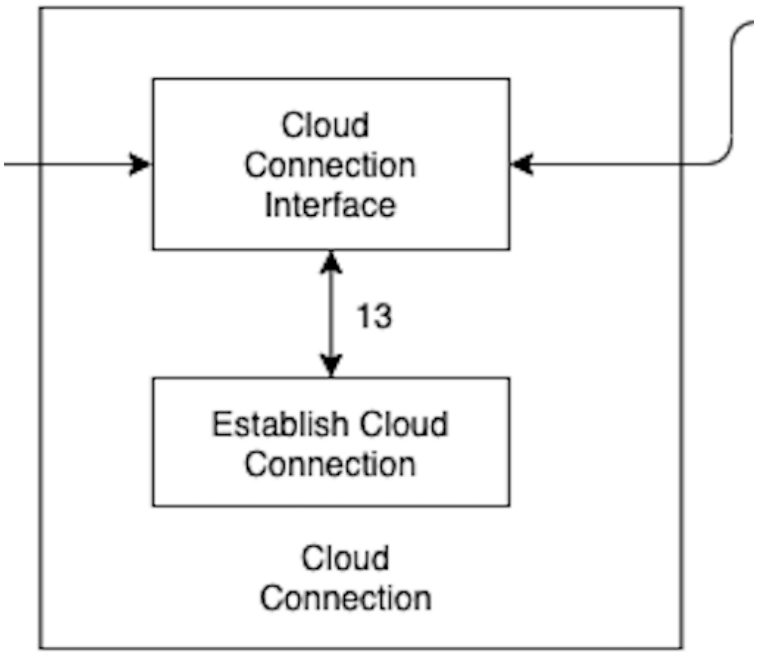



Figure 10: Cloud Connection subsystem description diagram

#### 4.4.2 Responsibilities

This subsystem is responsible for establishing a secure connection with the cloud and sending the data to the cloud.

#### 4.4.3 Data Processing

This subsystem uses the HTTP protocol to interact with the server on the cloud. The mobile app will send HTTP POST requests containing the location data, or messages to be sent to subscribed locations. The app will receive HTTP responses verifying everything is sent correctly.

### 4.5 GPS

In this section, the GPS equipment will detect and collect user position information and send back to the application which runs on userâs phone. The GPS will be collected using software provided by iOS.

#### 4.5.1 Assumptions

GPS interface receive application requirements to be able to start to track the user position then send back location data to the application

#### 4.5.2 Responsibilities

Application wake up the GPS tracker which the equipment embed inside userâs phone then start processing the location data and send back to the application

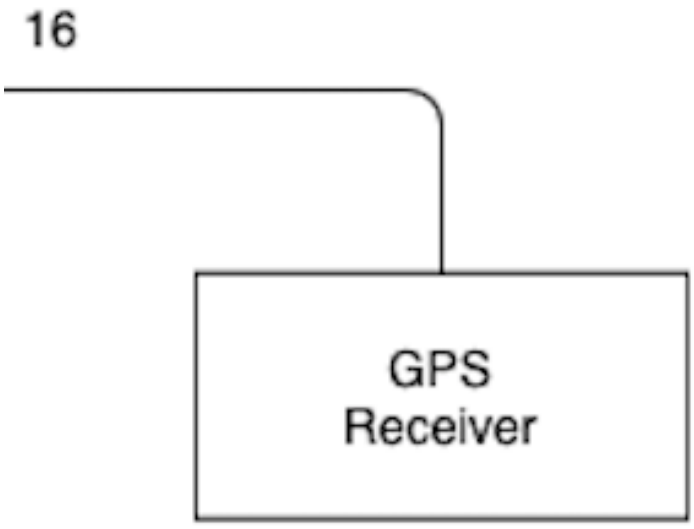



Figure 11: GPS Subsystem diagram

### 4.5.3 Software Dependencies

The library needed for GPS locations is CLLocationManager

## 4.6 BLE Detection

In this section, the BLE equipment will detect and collected by the user phone. The data will then be sent to be process to create a packet for tranmission.

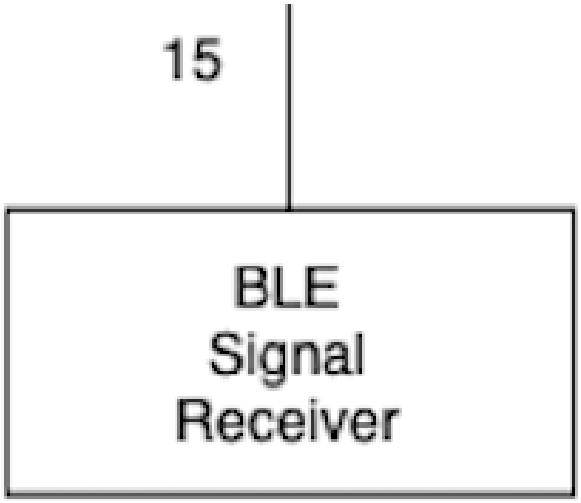



Figure 12: Example subsystem description diagram

#### 4.6.1 Assumptions

BLE interface send the broadcast signal wait for phone to join the BLE equipment and send back location data to the application

#### 4.6.2 Responsibilities

BLE equipment search for any available user in certain area and detect the user position then send back the information to the application

#### 4.6.3 Hardware

Estimote BLE beacons

#### 4.6.4 Software Dependencies

The library needed for BLE beacons is CLBeaconRegion

## REFERENCES