

**TOM-POI APP - High Level Design**  
**Revision Number: 1.0**  
**Flag-Ship Project - A1**

**Last date of revision: 6/8/2020**  
**Yogesh Badgujar**

**Change Record/History**

Revision	Date	Author	Changes
1.0	6/8/2020	Yogesh Badgujar	Initial Draft

**References / Particulars -**

<b>Client</b>	Tom Tom India team	
<b>Reviewed Team</b>	Tom Tom India team	
<b>BA</b>	Tom Tom India team	
<b>Architect</b>	Yogesh	
<b>Review and Approval By</b>	Tom Tom India team	
<b>Status</b>	Under review	
<b>IT Dept</b>		
<b>BU Head</b>		
<b>Manager</b>	Sapan Khandwala	
<b>SOW details/link</b>		

## **Contents**

### **1. Client Requirement (BR – 001)**

### **2. General Description**

#### **1. Objective**

#### **2. Assumptions**

### **3. Design Details**

#### **1. Main Design Features**

#### **2. Application Architect**

### **4. Application Use cases**

#### **1. CS Application**

#### **2. CS Assumptions**

#### **3. CS (Client) Use cases**

#### **4. SS (Receiver) Use cases**

### **5. CS Application Architecture**

#### **1. Architect design**

#### **2. Tools**

#### **3. Pattern**

#### **4. DB**

### **6. SS Application Architecture**

#### **1. Architect design**

#### **2. Tools**

#### **3. Pattern**

#### **4. DB**

### **7. [POC – Check the code on git](#)**

## 1. Client Requirement (BR - 001)

Point of interest data from moving Cars

Design a system to handle POI data transmitted from moving cars - Cars are able to capture image of surrounding, quickly recognize image, encode POI data into some format & send it over the air to cloud

-Check if the collected packets contain POI name & a valid category - rule based validation

-Persist them for downstream systems to consume

Aspects to consider :

Handle high volume with low latency

monitoring and reporting

Ability to plugin validations based on different categories.

## 2. General Description

1. **Objective** – Main purpose of application is to capture particular location/site (POI) data while traveling by car. What are those particular location? Those location are nothing but the nearby/roadside hotel, building, petrol pump, gym, movies theaters, hospitals etc. These data are in a form of photos.

Application can we use for security purpose like high profile people security, restriction area movement etc. Can use in Military operation using drone camera, military vehicle etc.

2. **Assumptions** – Application required hardware as well as software to build this application.

## 3. Design Details

### 1. Main Design Features

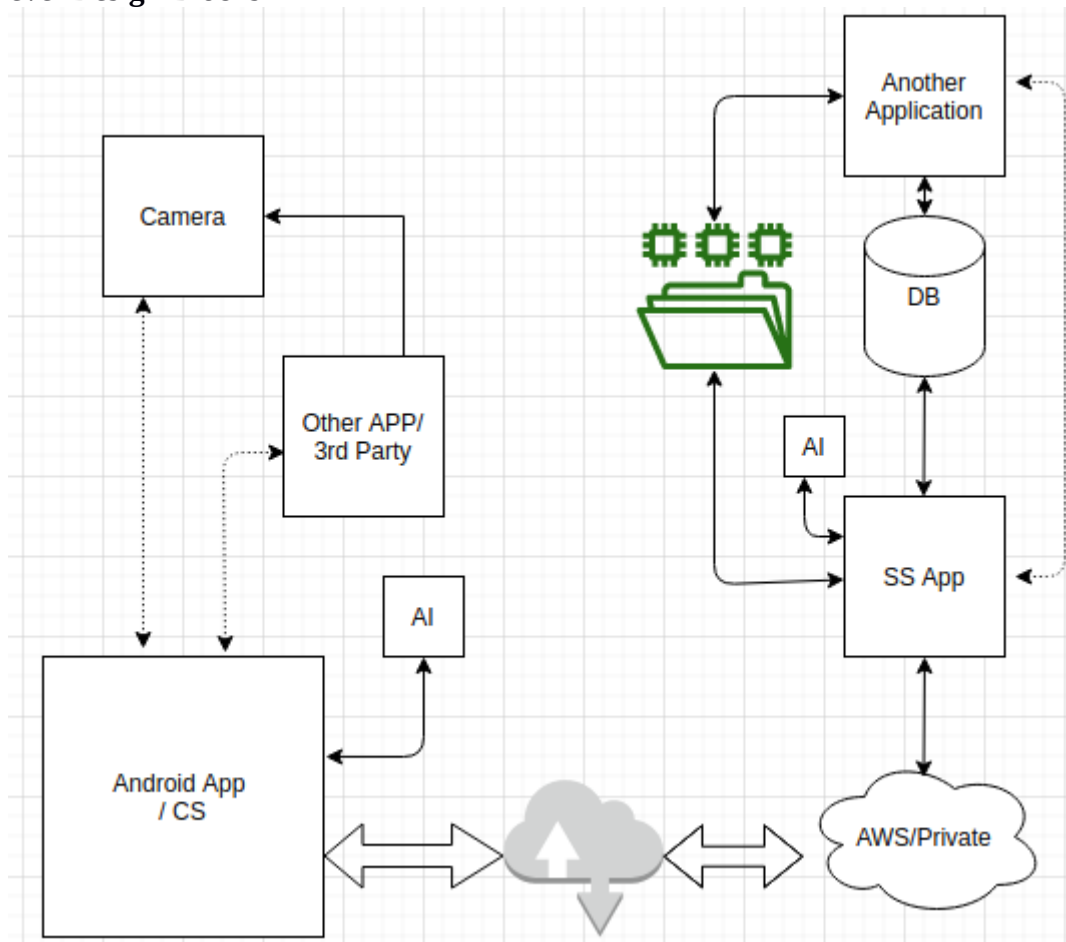
We need to build two applications, which will solve the current requirement as well as to cater the future upcoming requirements. Applications are Client side application(further will refer as Sender/CS) and server side application (further will refer as Receiver/SS).

### 2. Application Architecture

CS application will be the combination of hardware and software. CS application will be act as

sender. SS application will be completely on software base. SS application will be act as receiver.

### High Level Design Blocks -



### 3. Application Use cases

CS application can be build into five ways, along with deployment of the CS deployment part. Note application can be build in many ways.

**A) Mobile Application integration with 3<sup>rd</sup> party(Hybrid way)** – In this approach Our APP(which is android as well as IOS base) need to choose 3<sup>rd</sup> party service provider which has expose API to access camera based on your certain conditions (**Photos/ Images**) such as

1. Based on request it will capture the image .
2. Access to Image directory / configurable path of the captured images.
3. Should enable the GPS so that we can extract the latitude and longitude from the image as well at our end(SS application) to get more correct match(such as Exif, Gao-tages, Store location on photos etc).
4. Access to camera setting

In this approach need to type with or need to do license with other 3<sup>rd</sup> party system.

**B) Complete Mobile App will be develop** – Complete App will communicate with hardware such

as camera, deal with other mobile hardware, security and application etc.

**C) Complete App and integration with car camera and blue tooth/wifi etc** – Once application started or running at background need to integration with car front and back camera with blue thooth, wifi etc. Will be configurable at the time of application install or further configurable.

**D) Special hardware with other application** – In this case no need to depend on 3<sup>rd</sup> party, hardware etc. Set of 4 camera along with self device where the application will control and integration with the SS application using different techniques such as own GPS system, wifi or blue tooth techniques.

**E) Special hardware and application connect with user peripheral devices** – In this case instrument will only have processing unit for captured images and processing them and send to cloud system. External peripheral devices can be use for taking the inputs from user.

### 3.1 CS Application

There are different advantage and dis-advantages for each approach. Initial draft we are exploring the the different approach for CS application and do the POC.

### 3.2 CS Assumptions

As the core application development point of view, need to build application and do integration with 3<sup>rd</sup> party or build own device depending on suitable conditions and further funding/license cost etc.

As per initial POC with will consider following points for CS application

- Input to the application based on file system / gallery.
- There will be fix token base authentication with SS application

### 3.3. CS (Client) Use cases

We create Microservice based application that will be latter consider as base consideration / guideline for the CS application.

Application will be support in two ways **on-line** and **off-line mode**.

High level details or use cases for **on-line mode** as below

1. User registration at time of installation/start first time – Device will be register with the back-end application.
2. CS will send latitude, longitude, token etc based on that SS will give future/ near by / radius POI coordinates.
3. Matching of the latitude, longitude with current location/closer to the POI points, photos will be taken and kept at configurable location.
4. Once the photos kept at configurable location, will pick up image and following steps will be done.
  1. AI(Google Vision AI) will get the call to validate the photos.
  2. Labeling etc will be done on the image.
  3. Image will upload to SS application
  4. After successful confirmation image will be deleted from the location.
  5. POI coordinates will be store into DB along with few details (light weight DB).

High level details or use cases for **off-line mode** as below, most of the steps will be same only few steps will be develop for off-line mode.

1. Addition to Step 1 – also ask user for preferred start to end path for frequent way. Later user can configure the same as well. Based on path POI will be save to local DB.
2. Addition to step 2 – Through GPS you will get the coordinates and get the POI points or Even GPS system not up then need to take POI from History and do the further processing.
3. No changes in step 3
4. Step 4 will come in picture once switch back to on-line mode.

**Note** – CS application will rewrite depending on which approach has been selected. POS will get the better idea how the things will or move further.

### 3.4 SS (Receiver) Use cases

We create Microservice based application that will be latter consider as base consideration / guideline for the SS application.

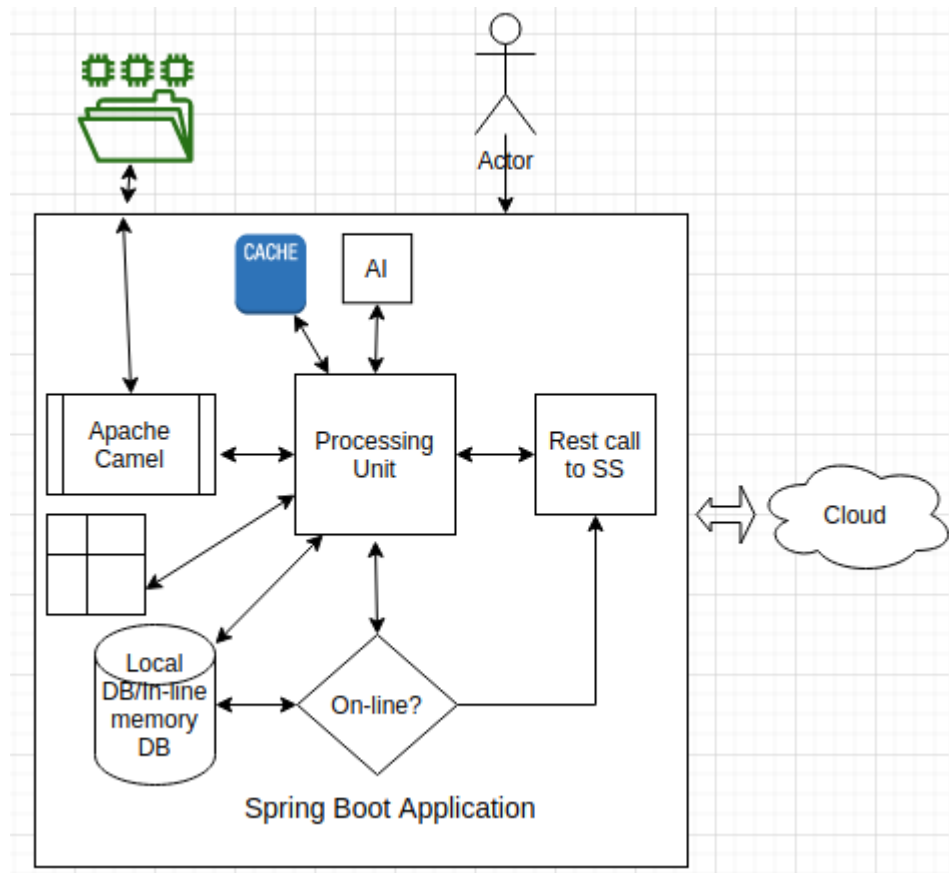
High level details or use cases as below

1. Different API will be created to interact with CS application. API are
  1. getPOI(POIDetails details) – get the POI details based on latitude, longitude, user auth details, radius, rough etc
  2. uploadPOI() - Upload the images to S3 or configurable DB.
  - 3.
2. Monitoring and reporting tools get integrated.
3. cleanUP() - remove of DB data based on server configuration/setting
4. weeklySyncDB() - push the data based on pattern or any analysis etc

SS application can we scale up or deployment can be done AWS or any other cloud environment.

## 4. CS Application Architecture

### 1. Architect design

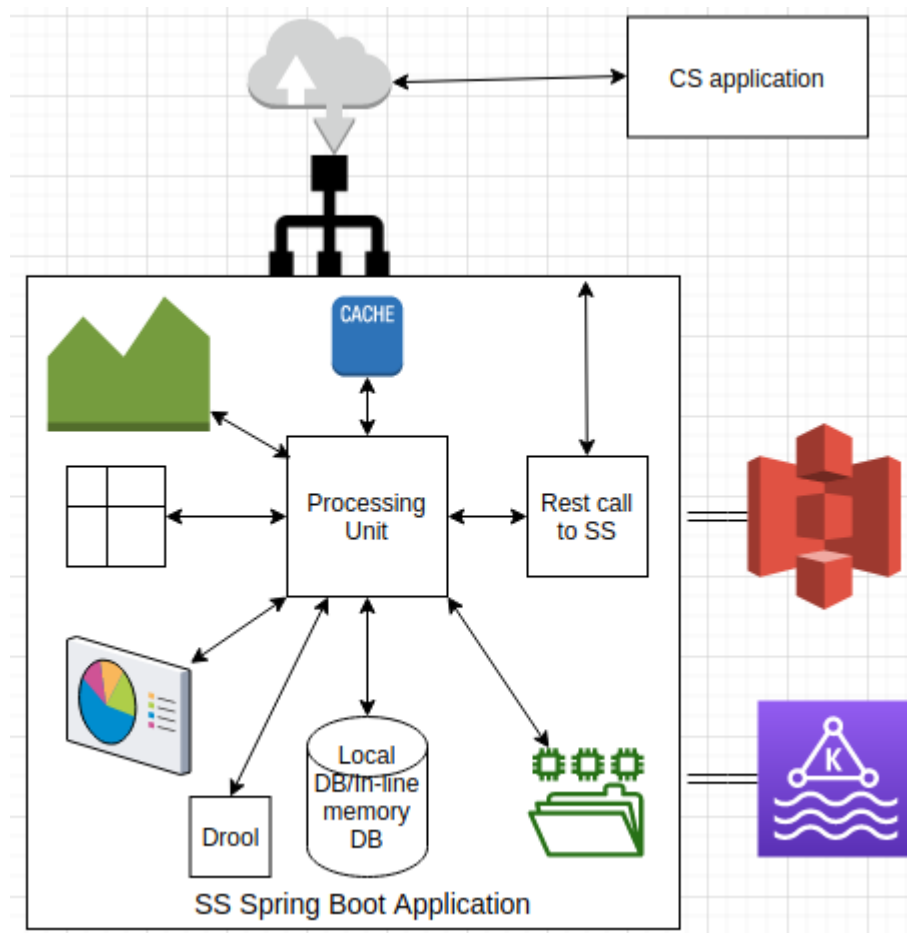


2. **Tools** – Java 8, Spring Boot 2.2, Google Vision AI, Apache Camel, Spring (module MVC, JPA, Scheduling), Mysql (with Spatial Data Types)/H2, Junit 5, Git, shell script, Swagger for testing/ Postman, Drool
3. **Pattern** – MVC pattern along with Hibernate - JPA so that in future replace with other DB.
4. **DB – No Sharding** required at CS side. Mysql/In-memory data will be used.
5. **Monitoring and Reporting** – Don't see much need at client side application. You can still configurable using Spring Boot Admin.
- 6.

## 5. SS Application Architecture

### 1. Architect design





2. **Tools** – Java 8, Spring Boot 2.2/ Akka, Google Vision AI, Spring (module MVC, JPA), Mysql (with Spatial Data Types), Junit 5, Git, Drool, AWS with different services based on need, Swagger for testing/ Postman
3. **Pattern** – MVC pattern along with Hibernate - JPA so that in future we can replace with other DB easily.
4. **DB** – Sharding MySql can be done on base on latitude, longitude. Initially it will be big efforts but in future retrieval time will be faster.
5. **Monitoring and Reporting** – Actuator for Monitor application(Spring Boot Admin).

6.

**6. POC – Check the code on git**

<https://github.com/YogeshBadgujar-javarefresh/POI-Application>

POI Monitor Application - <https://github.com/YogeshBadgujar-javarefresh/POI-Application/tree/master/POIMonitorAPP>

POI Server Application - <https://github.com/YogeshBadgujar-javarefresh/POI-Application/tree/master/POIReceiverApp>

POI Client Application - <https://github.com/YogeshBadgujar-javarefresh/POI-Application/tree/master/POIClientApp>