**Book A Cab – Design Document**

**Use Cases:**

There are two primary use cases in this project

1. Drivers send their location updates
2. When Customers book a cab, best driver is to be chosen among the nearby drivers



**Assumptions:**

1. There are two types of apps.

* A Driver app that sends the location of the driver continuously in fixed intervals(For us its 1 minute)
* A Customer app that requests for Cab giving a booking mode

1. There are two types of bookings possible.

* A shared booking mode (for simplicity we say 2 people can share a Cab)
* A single booking mode

1. There are two types of cabs

* Sedans
* Hatchbacks

**Service Division**

1. A Driver Location Service (DLS) that gets the data of drivers continuously in fixed intervals

* It pushes the location information to a Distributed Queue. For us its Rabbitmq
* The consumer of Rabbitmq is ElasticSearch, where location data of drivers is indexed

1. A Booking Service (BS) that takes customer location and gives back the best cab to the customer

* It also updates elastic search that a particular cab is booked after it gets the result from Careem Core Engine(CCE)
* It searches for all the available cabs near to the customer

1. A Careem Core Engine(CCE) that takes customer location from Booking Service and has its own logic of finding the best possible cab for the customer

* Based on the rating of the driver
* Based on elasticsearch results
* Based on the booking type

1. A Notification Service(NS) that can send the booked cabs current location to the customer App



**APIs:**

1. API to get Driver location updates

* /v1/location/{CabId}
* PUT
* Request Body :
* {
* id : <id>,

lat: <lat>,

long: <long>,

time: <yyyy-MM-dd HH:mm:ss>

}

* It’s a fire and forget call

1. API to book a cab by customer

* /v1/book/{customerId}
* POST
* Request Body:
* {

lat: <lat>,

long: <long>,

mode: <single/sharing>,

dest: <dest>

}

* Response Body:
* {

cab\_id: <cabId>,

driver\_contact: <driver mobile number>,

cab\_number: <cabNumber>

cab\_lat: <cab latitude>,

cab\_long: <cab long>,

driver\_name: <driver name>,

driver\_rating: <driver rating>

}

1. API to show all the cabs nearby to the customer

* /v1/search ? lat = <some lat> & long = <some long> & mode = <single/sharing>&type=<sedan/hatchback>
* GET
* Response Body
* {
* [

{

cab\_id: <cab id1>,

lat: <cab1 latitude>,

long: <cab1 long>

},

{

cab\_id: <cab id2>,

lat: <cab2 latitude>,

long: <cab2 long>

},

}

1. A Notification Service like GCM(Google cloud messaging) that can update the customer on drivers location