**High level Architecture Design Diagram for the Location based Taxi Aggregator and Selector (Cloud)**

A diagram of a computer

Description automatically generated

**Demo Presentation:**

1. Delete existing DB if present
2. Run InitializeTaxiAggregatorDB.py to create DB, Collections and create 2dsphere index
3. Run RegisterTaxis.py to create 50 sample taxi data
4. Run RegisterCustomers.py to create 5 sample Customer data
5. Run TaxiLocationUpdateSimulator.py to send location data for different taxis
6. Use below request to show Ride find based on customer location. API - <https://udmqfurfie.execute-api.us-east-2.amazonaws.com/dev/ride/find>

{

"location": {

"type": "Point",

"coordinates": [77.23149, 28.61123]

},

"type": "Luxury"

}

1. Post below json request to create a taxi data using API - <https://udmqfurfie.execute-api.us-east-2.amazonaws.com/dev/taxi>

{

"name": "Taxi\_051",

"type": "Basic",

"location": {

"type": "Point",

"coordinates": [

77.3368142118244,

28.66077927262116

]

}

}

1. Post below json request to create a customer data using API - <https://udmqfurfie.execute-api.us-east-2.amazonaws.com/dev/customer>

{

"name": "Manju",

"location": {

"type": "Point",

"coordinates": [

77.24098,

28.65454

]

}

}