Book a ride

We have 2 kinds of users in our application- captain and a customer, with data of the details of the user and the location. We are creating APIs that will listen to these application. And the service layer has 3 components :

Dispatcher system: This component is responsible for find the cab from the set of cabs available in the area, of the customer on the basis of a dispatcher algorithm and the rating of the captain.

Payments: This component is responsible for the billing of the ride, it calculates the distance/time and stores them. Also helps the user track all the rides previously taken by the user

Rewards and reviews: This component is responsible for the feedback of the ride and also helps match a captains repute for a particular user/ride.

Since we have a model where we are continuously looping for the coordinates of the captain, hence we have a messaging queue for asynchronous consumption of the location coordinates

The connector is responsible for the conversion of the messages to store it in Redis DB. We are making use of Redis DB for storing realtime data of the user and the captain.

We are making use of MySQL Db for storing persistent data such as user/captain details. And other billing information.

Architecture diagram:

MySQL DB

Connectors

REDIS DB

KAFKA Bus

Customer Application

Captain Application

Restful APIs

Payments

Rewards and Review

Dispatcher System

Booking System

(Connection Pooling)