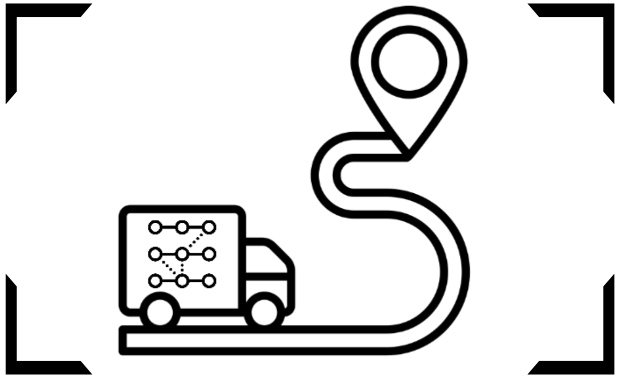
Route Optimization and Visualization for Sales Vehicles



MISHMASH

INDIA'S LARGEST ONLINE DIVERSITY HACKATHON

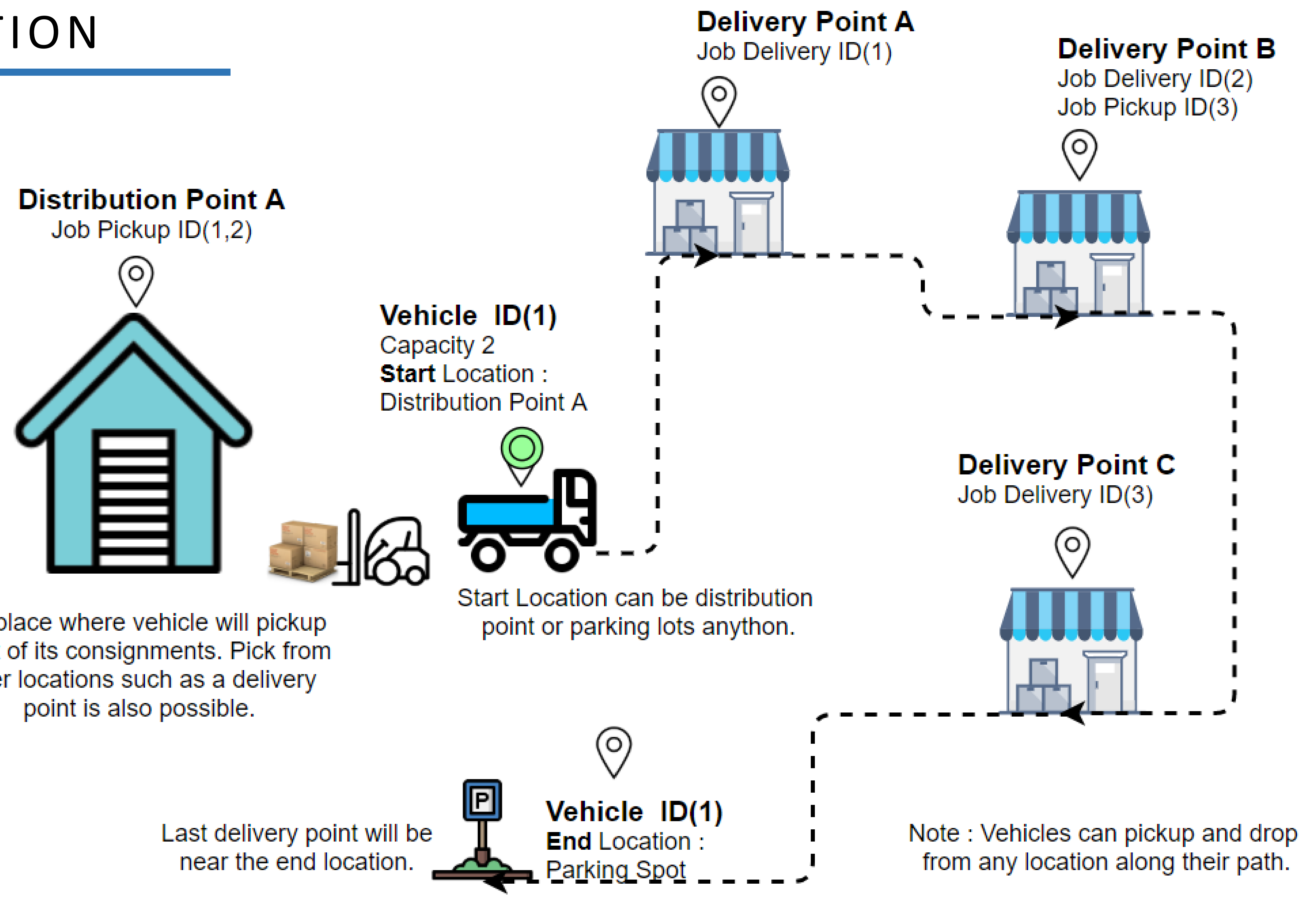
MishMash - India's biggest diversity Online Hackathon

Team
MERAKI

Kshitija Surange
IIT Kharagpur

Utsav Mishra
IIT Kharagpur

IDEATION



The place where vehicle will pickup most of its consignments. Pick from other locations such as a delivery point is also possible.

Start Location can be distribution point or parking lots anython.

Last delivery point will be near the end location.

Note : Vehicles can pickup and drop from any location along their path.

Vehicle Routing Problems

Vehicle routing problems (VRP) are essential in logistics. As the name suggests, vehicle routing problems come to exist when we have N vehicle to visit M nodes on any map.

We could say VRPs are a subset of **Traveling Salesman Problem** (TSP). In general, it looks like that:

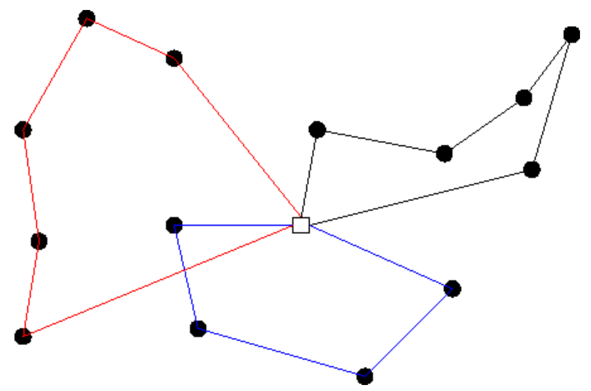
CVRP, VRPTW ⊆ SVRP, DVRP ⊆ VRP ⊆ TSP ⊆ Graph Theory

CVRP : If you have vehicles restricted by any capacity (e.g. max loading limit) constraint, then we call it Capacitated Vehicle Routing Problem (CVRP)

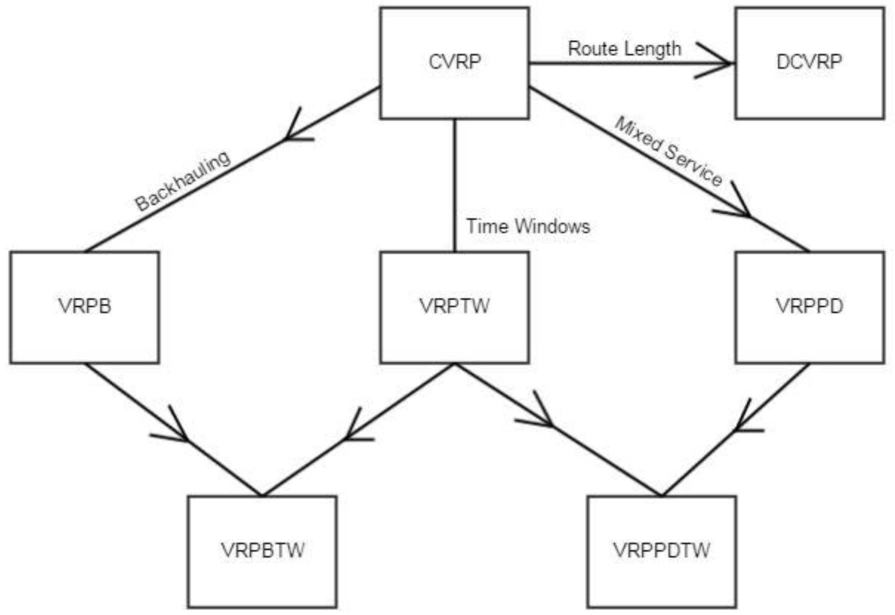
VRPTW : If you have vehicles restricted with working times, then we call it Vehicle Routing Problem with Time Windows (VRPTW)

SVRP : If the visiting points (nodes) are given, then we call it Static Vehicle Routing Problem

DVRP : If the visiting points (nodes) come to exist while trying to solve or moving on the map, then we call it Dynamic Vehicle Routing Problem

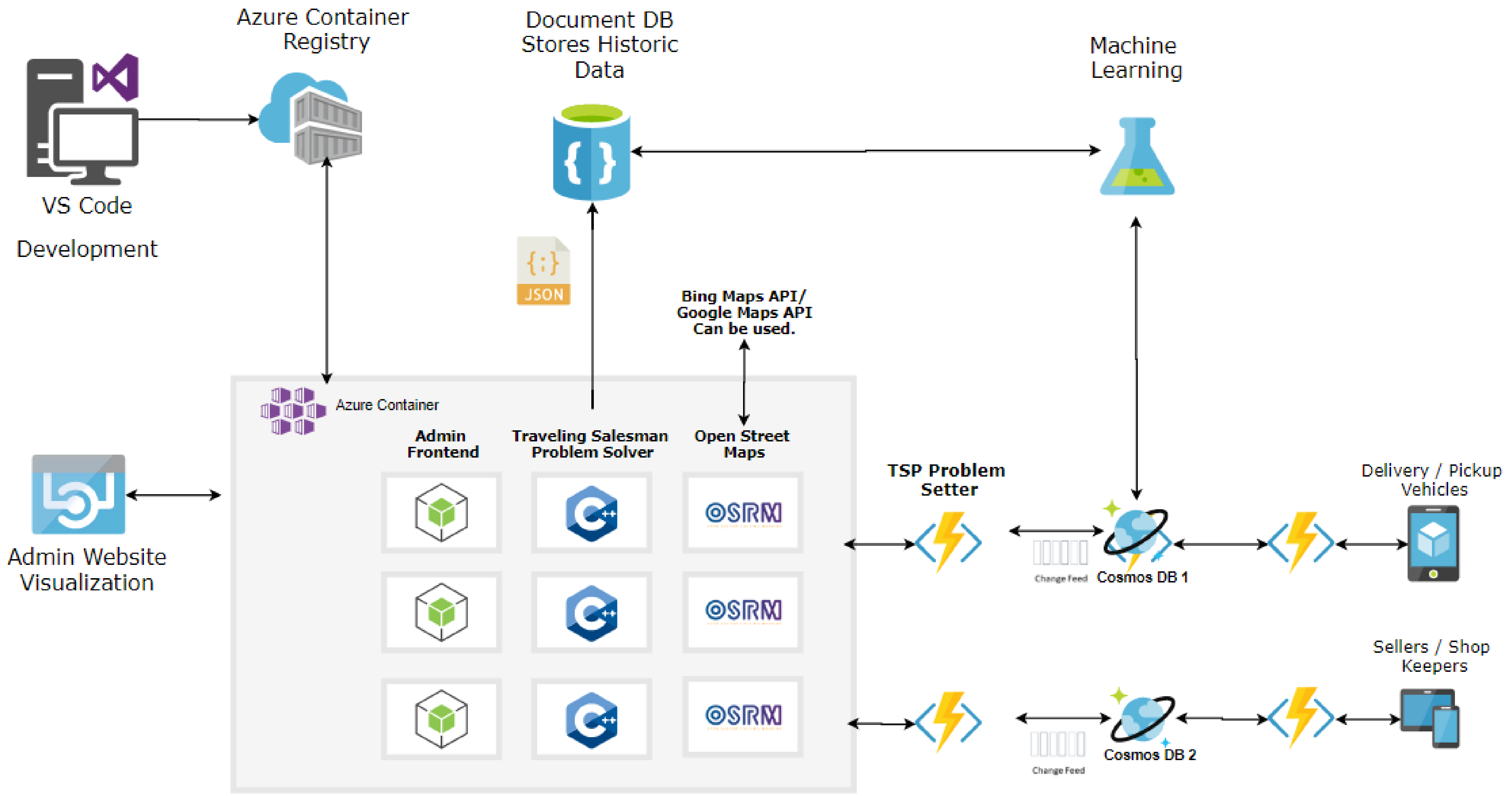


A figure illustrating the vehicle routing problem



Here is a map showing the relationship between common VRP sub-problems, from Wikipedia.

SERVERLESS-ARCHITECTURE (PROPOSED)



Note: This is our first server less architecture that we have attempted on Azure, feel free to punch holes. ☺

Tech Stack

PS Component	Tech used – Powered by the goodness of open source	Alternate Tech
Travelling Salesman Problem (TSP) Solver	• Project Vroom (Open Source, BD-2 License)	• Google OR Tools (Open Source, Apache License 2.0) • Jsprit by Graphhopper (Open Source, Apache License 2.0)
Distance Matrix API	• Open Street Routing Machine (Open Source, BSD 2-Clause Simplified License)	• Distance Matrix API by Google Maps • Bing Maps Distance Martix API
Visualization	• Open Street Maps & Project Vroom (Open source, Open Data Commons Open Database License)	• Mapbox • Streamlit • Google Maps • PowerBI
Deployment	• Docker	• Azure Container Registry • Azure Kubernetes Service

SOLUTION – Assumptions On Input Data

Delivery Vehicle:

Operated between Distribution Point and last mile Delivery Points, having vehicle Identifier (num/alphanum).

- **Multiple types of vehicles** :- Bicycle, Motorcars(Mini-Trucks), Heavy goods vehicle, Walking / Jogging
- **Speed** : Indeed a more than complex parameter to decide (Check Annexure 1), until its cycle vs. motor vehicle.
- **Cost** : Indeed cost is an important feature, which **can be** included in a **custom matrix** implementation. Currently I have assumed all vehicle cost as same.
- **Supply-demand balancing**: Multiple Pickup and Drops along the path, this means apart from picking up the consignment from the Distribution point, a vehicle can pick up smaller consignments along the way if it has capacity in a trip.
- **Location** : Necessary either Start (Lat,Long) or End (Lat,Long). If both are present then the vehicle starts and ends from the given location.
- **Skills :** Basically Mapping of different types:
  - Product-Vehicle : Cold Storage Vehicles can only Transport a particular product
  - City – Distribution Point – Lat,Long : The algorithm can run for entire world at real-time, we need to make sure of some mappings in order to make it possible.
- **Capacity** : Capacity can be set per vehicle, capacity can be number of items, weight or volume, number of drops.
- **Working Hours** : Can be set so that a driver drives truck till 2 PM, have Lunch and then Drive or (9am to 6 pm).
- **Real-time Allocation** :- Real-time Location if Present that Jobs/Tasks can be added to a the path of vehicle in near real-time, if the any available vehicle has a required capacity, or stock and pickup/drop is along the route.

SOLUTION – Assumptions On Input Data

Pick up or Delivery JOBs for vehicles :

Can schedule orders for delivery/pickup in Advance or Same day as per their requirement with Identifier (numb/ alpha-numb).

- **JOB Type** :-
  - Pick up (at distribution or delivery point),
  - Delivery (at distribution or delivery point),
  - Shipment (pickup and delivery same day-trip from a vehicle)

- **Location :-** Pick up or Delivery (Lat,Long) depending on Job, Pick up and drop in case of shipment.
- **Job Time** :- Time taken to complete the job. (Off Loading Material, Bill Payment, Packing)
- **Skills** :- Mapping constraints such:
  - Refrigerator truck can handle refrigerator, Vehicle Type – Job Mapping etc.
  - Skilled agent and driver team sells to a particular seller. Let **ML model** decide this as personal connect matters.

  Note : Skills of job will always be a subset of skills of vehicle in output allocation.

- **Items** :- Number and types of items to be picked or delivered, helps in keeping tack of capacity of vehicle.
- **Time windows** :- Say what between what time the activity has to close. Eg: [“10 Am to 1 Pm”, “3 Pm to 6 Pm”]
- **Priority** :- How important is the job, worst case can it be skipped for the day to save cost. Let **ML model** decide this. ☺

SOLUTION – Output Data

Output would be written as a json file.

Routes

- **Routes** : For a given fleet of travelling vehicles, it gives the route to drop-off/pickup(if any) delivery consignments step-wise for each vehicle that needs to follow and complete within the considered time duration.
- **Delivery** : The number of deliveries to be made by a vehicle in the considered time duration
- **Pickup** : A pickup of consignment, if any, which a vehicle can carry out along the route of its deliveries depending upon its capacity.
- **Cost** : Cost for the route, it is the time taken which the TSP solver tries to optimise
- **Service** : Total service time for the route
- **Distance** : Total route distance
- **Waiting Time** : Total waiting time for the routes

Summary

- **Unassigned** : Number of jobs that could not be served. The algorithm tries to optimise travel time, skips jobs in case it has less priority and is far away i.e, costly to go to.
- **Service** : Total service time for all routes
- **Duration**: Total travel time for all routes
- **Cost** : cost for all the routes
- **Waiting Time** : total waiting time for all routes
- **Distance** : total distance for all routes
- **Delivery** : Total delivery for all routes
- **Pickup** : Total pickup for all routes

Steps

- **Type** : Job description – pickup, delivery, start job, end job
- **Arrival** : Estimated time of arrival at this step
- **Duration** : Cumulated travel time upon arrival at this step
- **Location** : Lat-long coordinates of the step in the route
- **Load** : Vehicle load after step completion considering capacity constraints
- **Waiting time** : Waiting time upon arrival at this step

SOLUTION – Key Takeaways

1

The TSP solver algorithm tries to optimise travel time, skips jobs in case it has less priority or if the job is located is far away i.e. , the job is costly to be carried out.

2

We need to provide all the vehicles available to carry out the set jobs and all the jobs that are to be done in a considered time duration. The TSP Algorithm tries to find most cost optimized solution while trying to use minimum vehicles hence, maximizing their productivity while keeping check on the capacity of vehicle, and time window(working hours) of driver.

3

TSP Solver Implementation offers option to input custom matrix, which means that we can configure items like cost, in case we want to put average cost per km as the optimisation parameter rather than time taken.

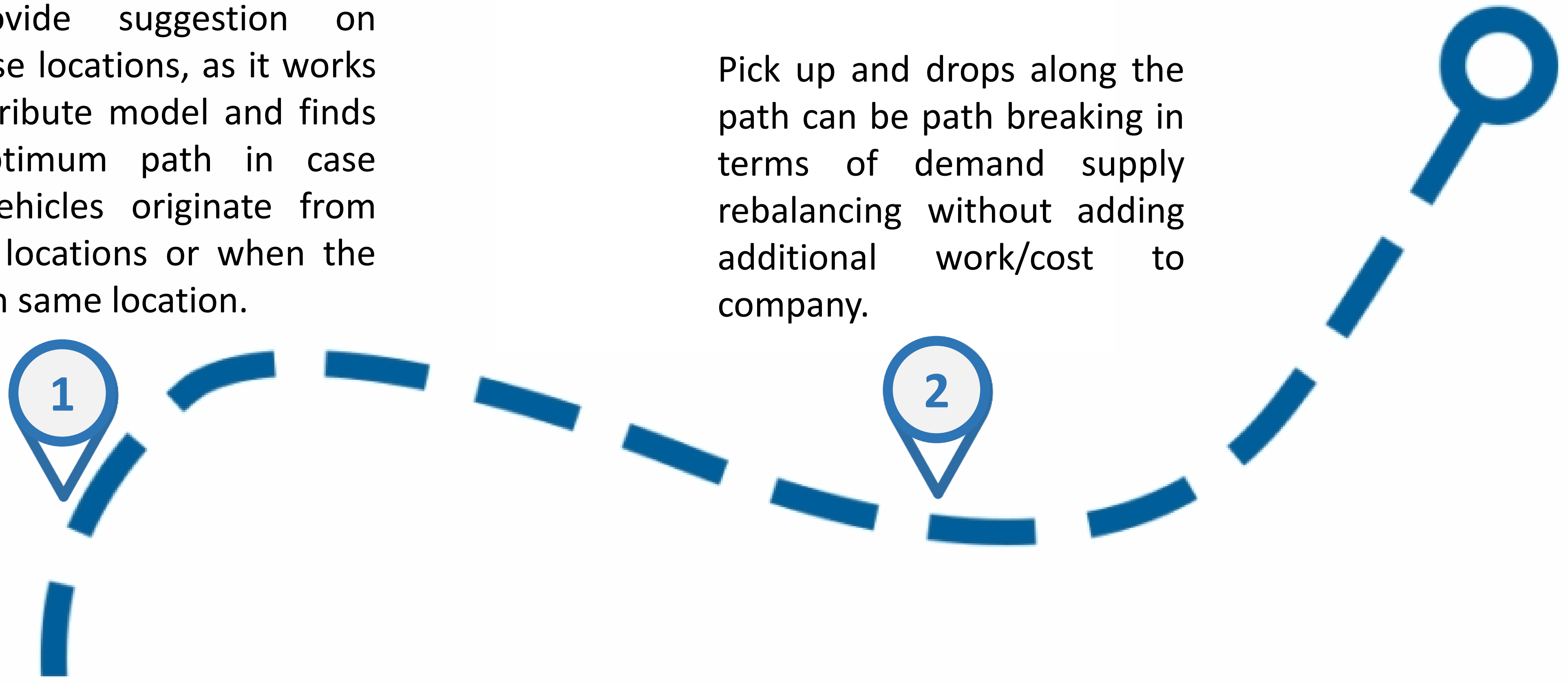
4

Priorities of order/delivery can be very easily generated from regression on historical sales data also what vehicle and agent visits which shop/sellers can be derived from a simple clustering followed by a classification ML model.

SOLUTION – Roadmap

Current implementation does not provide suggestion on warehouse locations, as it works on a distribute model and finds most optimum path in case where vehicles originate from different locations or when the start from same location.

Pick up and drops along the path can be path breaking in terms of demand supply rebalancing without adding additional work/cost to company.



ANNEXURE 1 – Speed of Vehicle

What Matters is road types and not Average/Max Speed.

- Speed Limits on Indian Roads (https://en.wikipedia.org/wiki/Speed_limits_in_India)
- Road Types Vs. Vehicle Types. (https://wiki.openstreetmap.org/wiki/OSM_tags_for_routing/Maxspeed#Motorcar)
- Speed can be fixed with the input data but it is not a good idea as for a street type road all vehicles will move with the same speed. But if we set a particular speed say 30 Kmph, it would valid only for one road type. Thus such seed data is tagged with roads on map data and OPEN STREET MAPS, GOOGLE MAPS, BING MAPS take care of it according to vehicle type and it is called **Average speed per way.**
- Until and unless we are comparing walking or cycle with motor vehicles, average speed does not matter. Open Street Maps gives us an option to define it our self.

Read the discussion below:

- https://wiki.openstreetmap.org/wiki/Talk:Average_speed_per_way
- https://wiki.openstreetmap.org/wiki/Proposed_features/Practical_maxspeed

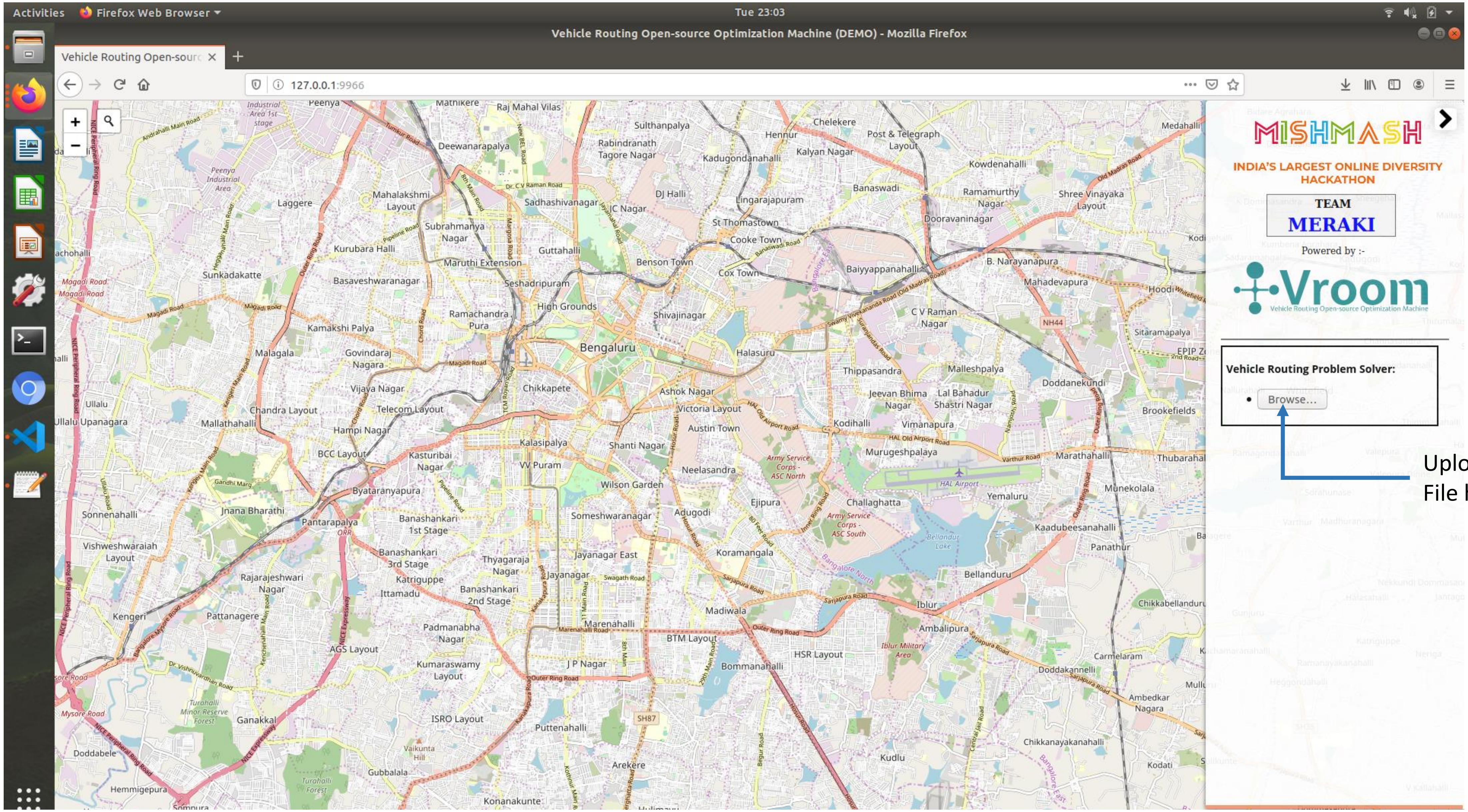
SERVER-LESS EVENT DRIVEN ARCHITECTURE Explained

Assumption: The application will work in real-time and accept request from shopkeepers, allocating their jobs to vehicles.

- **VS Code**:- For development and pushing docker images to Azure docker registry with Azure command line tools.
- **Azure Container Registry**: For maintaining the following images through development cycle.
  - Open Street Routing Machine (OSRM) – For map data, distance matrix. Self hosted OSRM saves network time & cost for POCs. Google Maps and Bing Maps have similar distance matrix API which can be used.
  - Traveling Salesman Problem Solver.
  - Frontend to visualize the output of TSP solver, built on node-express on Open Street Maps. Can be built in any
  - Map Visualization tool which plots polyline encoded route geometry or JSON Lat,Long.
- **Cosmos DB**: No SQL DB with support of change feed and secondary Index
  - DB 1 – Keeps Track of Available Vehicles, Keep track tasks status (delivery) updates.
    - PK- Vehicle Identifier, SK- Date (availability), Change Feed Enabled
  - DB 2 – Keeps track of sellers, requests for delivery, date of request, Last Visit and Sales.
    - PK- Store Identifier, SK- Date of Request/Last delivery, Change Feed Enabled.
- **Azure Functions**: State Less Functions to set travelling salesman problem and feed it to solver.
- **Azure Kubernetes Service** : For running docker containers.
- **Document DB**: To store historic as well as latest problem data, this data will be used to set priorities for seller request and make sales agent – product type mapping with Machine Learning.

Note: This is our first server less architecture that we have attempted on Azure, feel free to punch holes. ☺

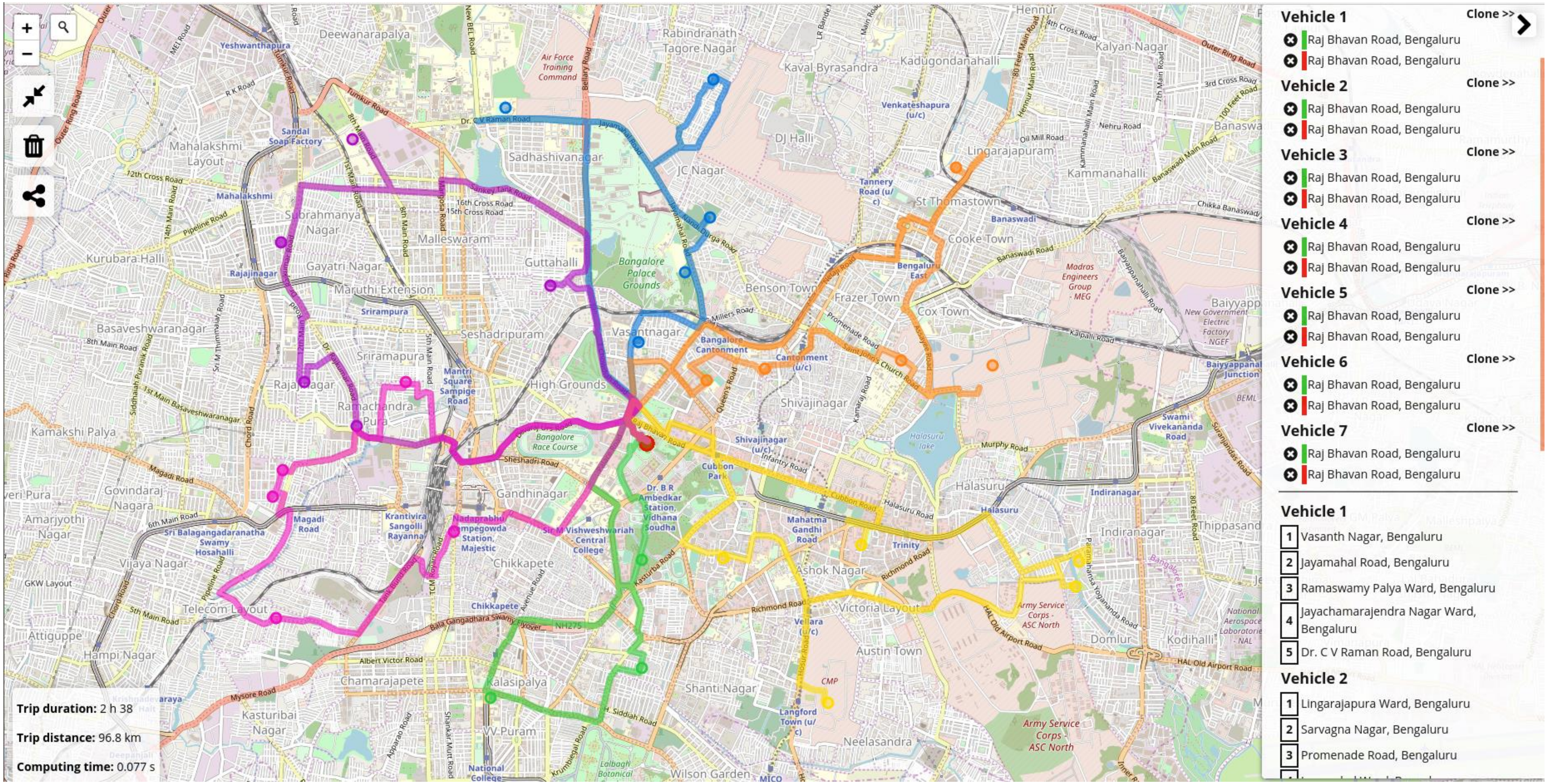
Front-end Demo


Activities
Firefox Web Browser
Tue 23:03
Vehicle Routing Open-source Optimization Machine (DEMO) - Mozilla Firefox
Vehicle Routing Open-sourc
127.0.0.1:9966
Bengaluru
MISHMASH
INDIA'S LARGEST ONLINE DIVERSITY HACKATHON
TEAM
MERAKI
Powered by :-
Vroom
Vehicle Routing Open-source Optimization Machine
Browse...

Vehicle Routing Problem Solver:

Upload JSON File here

Front-end Demo - Routes



Vehicle 1 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 2 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 3 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 4 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 5 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 6 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 7 Clone >>
- Raj Bhavan Road, Bengaluru
- Raj Bhavan Road, Bengaluru

Vehicle 1
1. Vasanth Nagar, Bengaluru
2. Jayamahal Road, Bengaluru
3. Ramaswamy Palya Ward, Bengaluru
4. Jayachamarajendra Nagar Ward, Bengaluru
5. Dr. C V Raman Road, Bengaluru

Vehicle 2
1. Lingarajapura Ward, Bengaluru
2. Sarvagna Nagar, Bengaluru
3. Promenade Road, Bengaluru

Trip duration: 2 h 38

Trip distance: 96.8 km

Computing time: 0.077 s

Thank you !

Team
MERAKI

Kshitija Surange

Utsav Mishra