

Be

EVRA

⇒ EVRA serves as a navigation function for fully electric vehicles. It is an online function and multiple route calculation is done in the cloud.

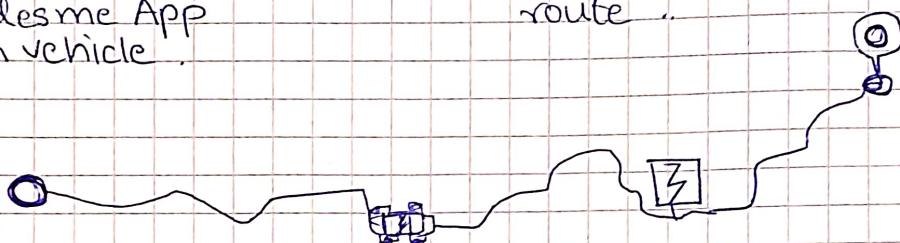
EQ optimized } always plan route considering
navigation

- * Dynamically changes :
take place via
→ Mercedes me App
→ within vehicle .

① shortest charging

time .

② Calculates fastest
route .

EVRA FEATURES :

- ① Don't choose CS (Charging station) near source (or) destination .
 - ② Keep enough distance between the CS .
 - ③ Utilization of battery (Good)
 - ④ Don't charge if battery level above 80% ("Avoid!")
- } IMPLEMENTATION OF SHORTEST PATH OF GIVEN WAY POINTS.

② CHARGING TIME DISTRIBUTION ALGORITHM

① Plots the route with best charging station .

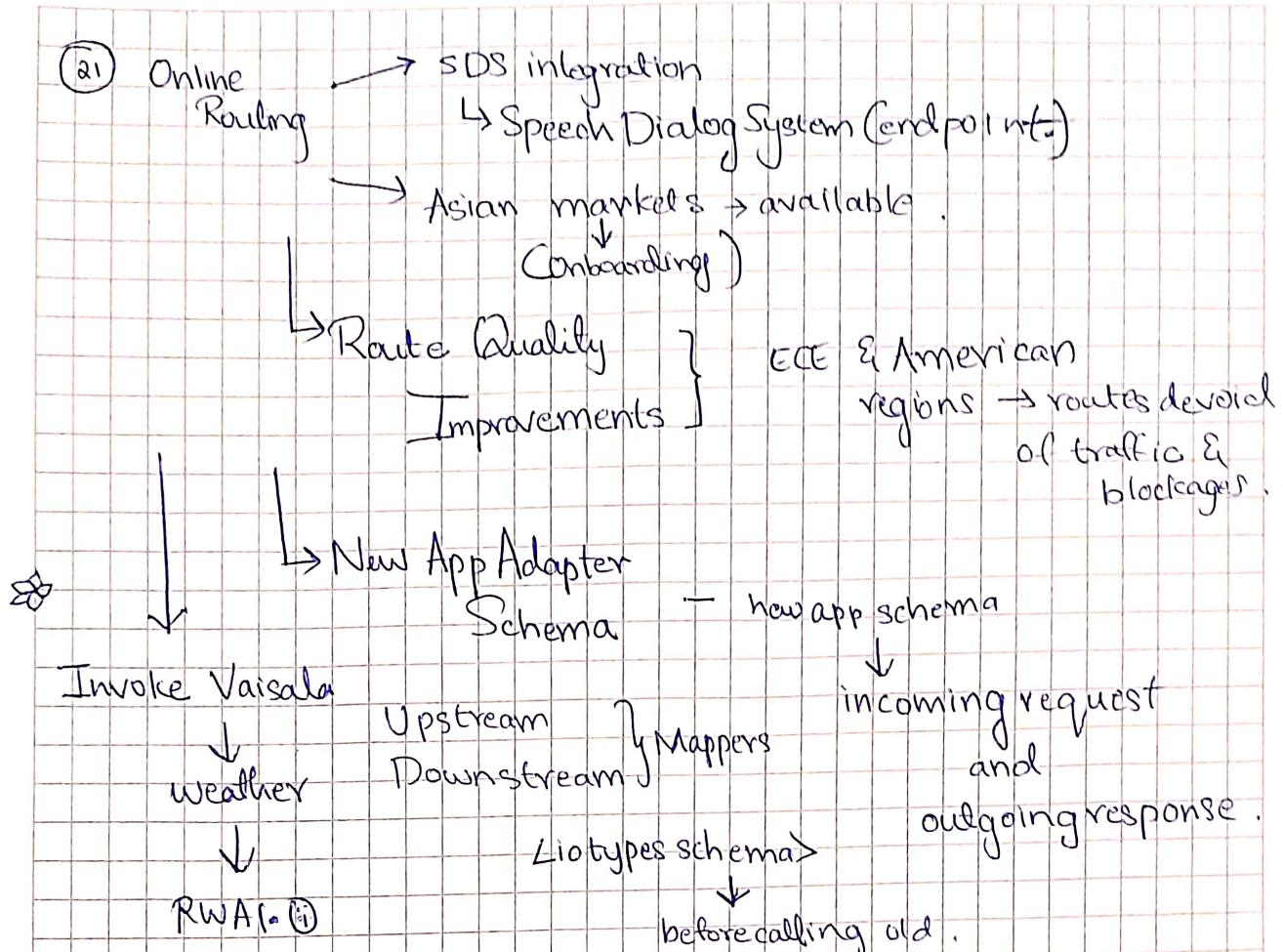
② Calculates duration / time based on power level of the CS !

IF > 3000 km
pops up a message and
alt route . (Max)

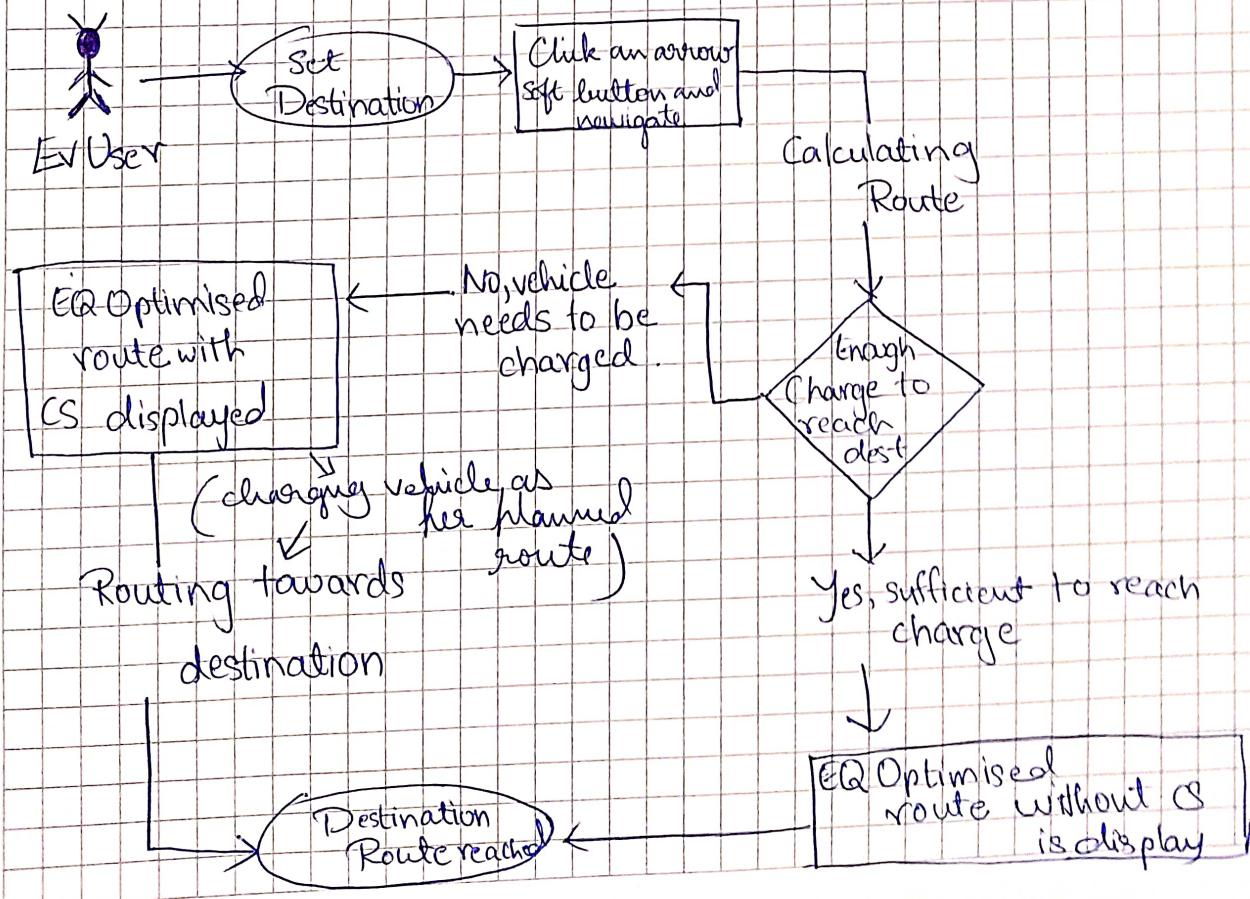
} III Beeline DISTANCE

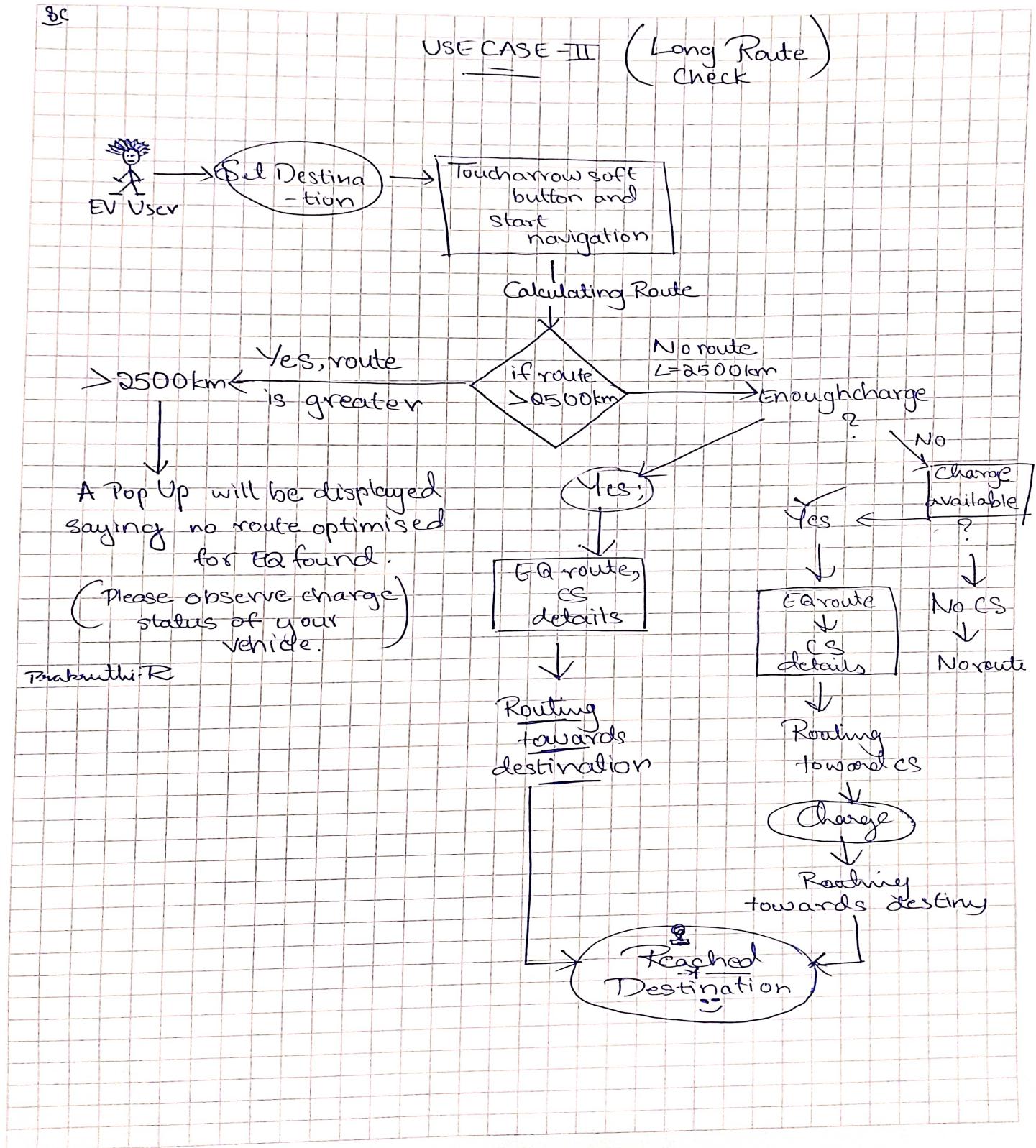
- (IV) Charging Station } Applies filter and
Filtering } checks for the available station .
- (V) Provide zero results } When selected route has more
than 8 charging stations
↓
pops msg → "Maximum CS limit exceeded"
- (VI) Ignore charging stations in the avoid area } ignore CS in avoid areas ; if no route is possible
↳ select from avoided area .
- (VII) User stopover } Implement user stop over in the route (user defined destination)
- (VIII) Charging duration } Charging Station energy
110 kW - minimum charging level is 10% .
≥ 110 kW - minimum charging level is 15% .
- (IX) Blacklisting of charging stations } avoid CS that user ignores or sets to avoid .
- (X) Send2Car - OVN } Send route from mobile to HU for oversized vehicles .
- (XI) Get rid of minimal charging } Minimal amount of charging to avoid (85% - 100%)
- (XII) Minimum Soc @ CS } 7%
- (XIII) Minimum Soc @ 10% est } 2% Destination

- (13) Manual Selection of CS } manually selected CS → returned as "planned charger" ↓ Charging Point.
lower charge power → returned.
- (14) Reduce AC Hops } if soc @ CS not min; fall to 10% to reach DC; avoid AC.
- (15) Charging stations } → Prefer on highways (while travelling on highways)
- (16) Charge @ Destination } suggest possible charging locations around cust end dest.
- (17) Hybrid Range Assist } provides route for hybrid vehicles.
- (18) Honeypot Feature } No active routes :- current battery status, } go to station.
- (19) Removal → Ac hops chargers.
Since density of DC less in AMAP
direct to AC charge to reach DC.
- (20) Ionity Service } ionity chargers are prioritized in the routes for customers having active Ionity Service.
* alternative routes added
* NTG6 adapter ↳ EVRA 2.0

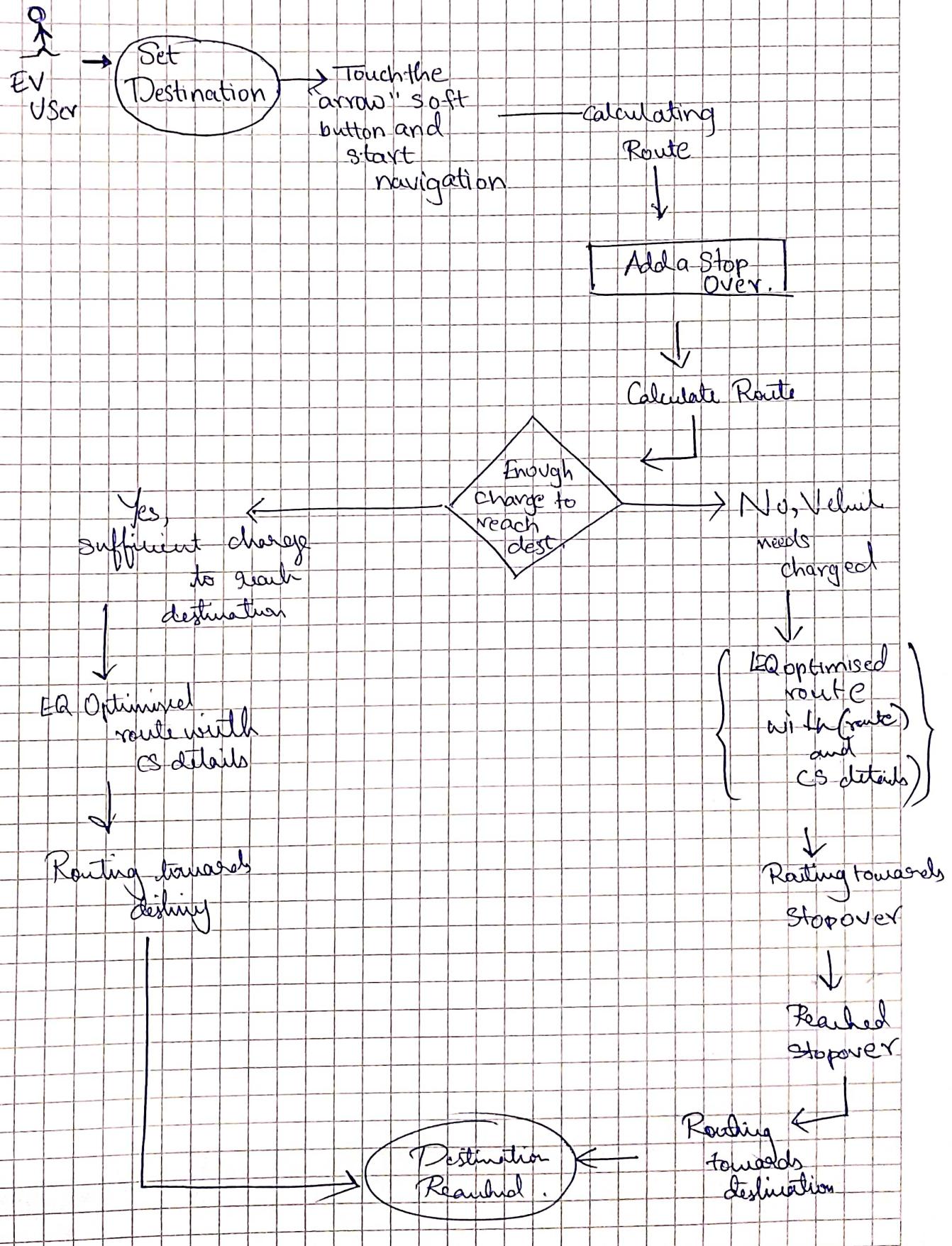


JSE CASE : I (Eq Optimised route)





USE CASE III } STOP OVER



* MFS Schema

config.ini

↳ endpoints

↳ client secrets

MFS Schema

↳ used by HRA Service

MFS → Routing

(Used by downstream)

① ts - timestamp

② hvBattery - battery related details.

↳ charging info, state

③ State → maxEnergy (w/sec)
curEnergy

attribute

→ vehicle

↳ trailers

plugged

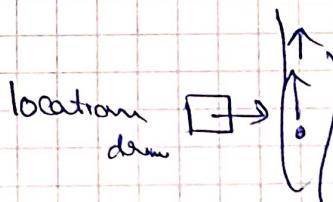
↳ contained at
back of vehicle :-

Calculate → $\frac{\text{current Energy}}{\text{max Energy}} \times 100$

= 20% healthy left

→ location

↳ lat
long



→ bearings

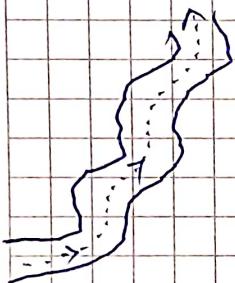
↳ angles of the car.

→ events - different events / changes.

- * whichever calls under map \hookrightarrow map features
 - * routes
 - \rightarrow initial request
 - onboard ID: ID given by Headunit
 - * waypoints
 - \hookrightarrow ID
 - onboard ID \rightarrow for destination
- stopover 1 \rightarrow 3
 (2 \leftarrow 1)
- geo coordinates detail for each waypoint in map features
- ** points coming along the way
- waypoint \rightarrow request.

\rightarrow priority \rightarrow zero / one.

① calculate new route increment by one



prior : 1 (request)

+ 1 = 2 (priority)

higher priority \rightarrow latest response

① battery

(battery status)

Current route

changed

② vehicle

(trailer plugged)

⑤ navigation, priority

destination

\hookrightarrow map failure

③ fuel (tire)

\rightarrow avoid map point
charging ID

④ location :- $\begin{matrix} \text{low lat} \\ \text{low lon} \end{matrix}$ (origin)

⑥ new settings stat. Clearabit

be Only Routes Modes

id:

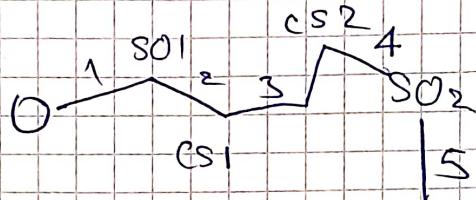
state: Search
start [true...]

occupancy

↳ no. of people.

client :- head unit
version .

language : -
1
2
3



5 segments
blw 2 way points
segment.

arrival
difference = delta

polyline
widly → segments
(polygons)
polyline

destination,
↳ routewaypoint

choose
polyline

ord, eqrg, eqr, mpp

8e

Routing Schema

event {

Knownroutes which were active in mfs
mapped to known.

clientID → routeclientID

hv → inside

use case data .

client ID

fixed
priority
for.

fixed-tree
client ID

type
"stopover"

*ARA-CAD

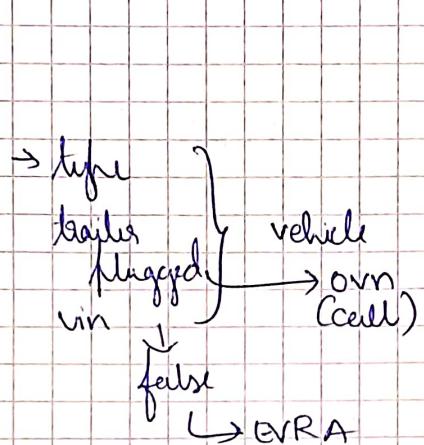
* App Adapters

- ↓
ORD (Online Routing Dispatcher)
- ↓
EARG (EQ optimised routing gateway)
- ↓
EQR (EQ optimised routing)

(1*) ARA : App Routing Adapter CAD : Charge At Destination.

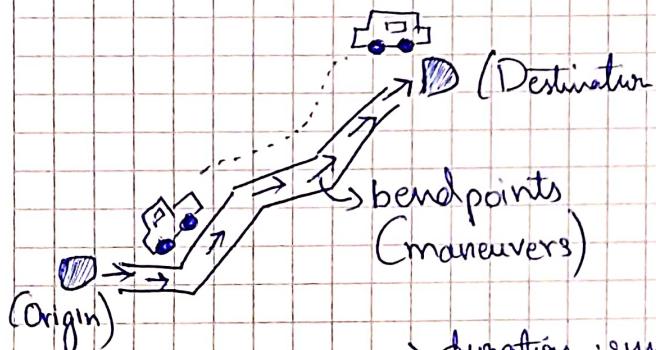
* Waypoints :-

- type :
- location { latitude }
→ longitude }
- id
- electric { soc
settings }



→ segments
legs

Charging station



→ duration : sum of duration of all legs.

$$\text{Length} = \frac{\text{origin to}}{\text{sum of charging station}} + \frac{\text{sum of CS to dest}}{\text{CS}}$$

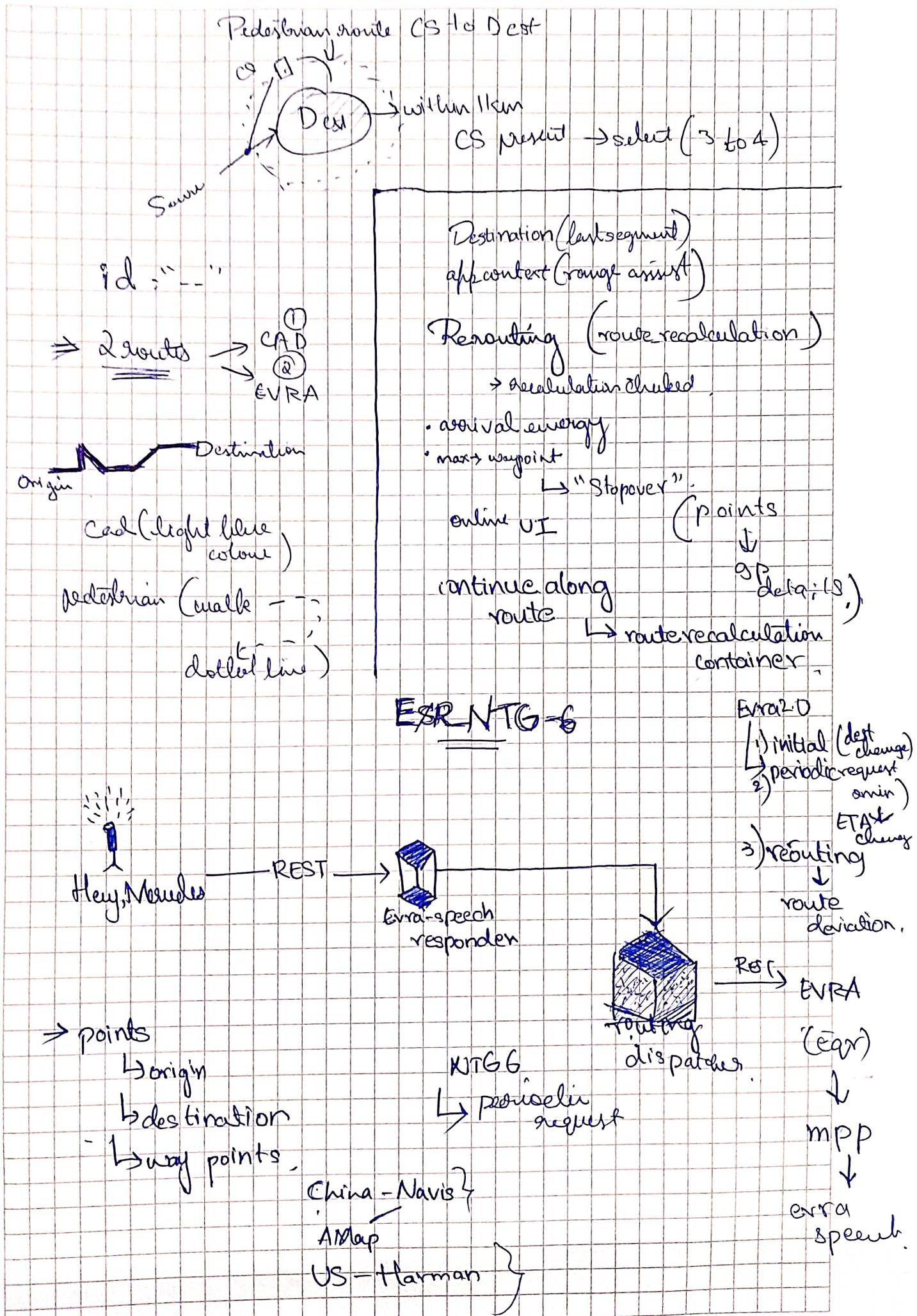


(2) CAD : Charge at Destination

CAD triggered when 5km away from Dest.

→ triggered by ORD (ORD checks distance)

Charge at dest service.



★ EVRA_FRAMEWORK_WALKTHROUGH

→ Erra1.0 → xRAY

Erra2.0 → Confluence,

(online automation side).

go run & cmd/map.../main.go

sh .\Scripts\local-setup.sh

→ io.asserted-port

→ cucumber

↳ write test cases

Config (endpoints)

★ INFRA-KT

④ Our services → AKS Kubernetes service.

ORP
ORP
RWA

clusters

Kubernetes



contains
orchestration

deploy

nodes ← node (smallest unit)
(VMs) ↓

cluster

(manage F1KS)

Azure : deploy into
subscription

Total : 21 subscriptions.

Binan } 5 (RWA)

} EVRA1.0

OFF BRT (5 -EVRA1.0)

EVRA
2.0

{ ONLINE
ROUTES } 5

nodes → virtual
machine

China } 6

Total = 21

→ ec2-devtools-onlineroutes-less | node pools

⇒ check status of node pool
if there is a
crash

Key Vault → secrets . restart if not ready ,

Datadog : monitoring dashboard

↳ check latency errors
performance issue

Opsgenie → alerts
(incases)

→ Services , traces
dependency map (upstream)

fix issue

Lens : deployments
(Unix command)