Requirements

See all nearby drivers

Customer req a driver

Once accepted share location for both

Drivers need to regularly share their current location

Non Functional Req - Availabiltiy, Scalability, Latency

Capacity Estimation

For Customers -> 500 Mn users and 1Mn DAU

For Drivers -> 1 Mn drivers and 500k DAU

Drivers share their location every 3 seconds, when
cab is booked tracking become realtime.

Driver(36 bytes for 1 driver) - DriverId(Long), OldLat(Long),
OldLat(Long),NewLat(Long),NewLat(Long)

1Mn * 36 bytes = 36 Mb , we have to store in 1 cache, but we will save it in different according to different countries.

Considerations

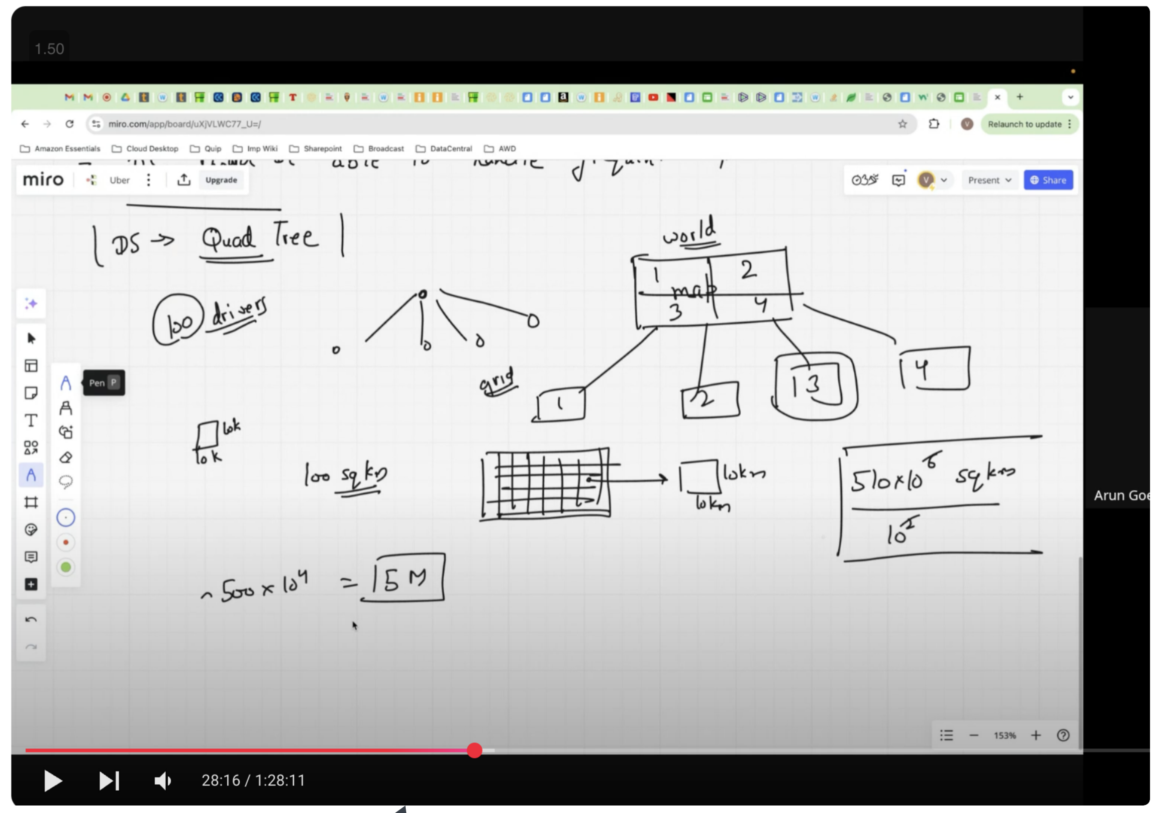
Design a Data Structure(DS) to handle geographics based on driver count.

DS should be able to handle frequent updates.

We can use QuadTree(4 childs) here.

One grid can have -> gridId, Latitude, Longitude, D

We can now find dirver location as - Selct * from grids where Lat between (X+y,X-y)
AND Long between (Y+d,Y-d)... but there can be many drivers in one grid and very few in some , so we use dynamic grid. Advantage is lower grid to find drivers, but disadvantage is dificult to update as location is constantly changing.



Design when we Open app and Drivers are moving

We can write through cache every 3 second and after 15 seconds we can update the quad data structure.

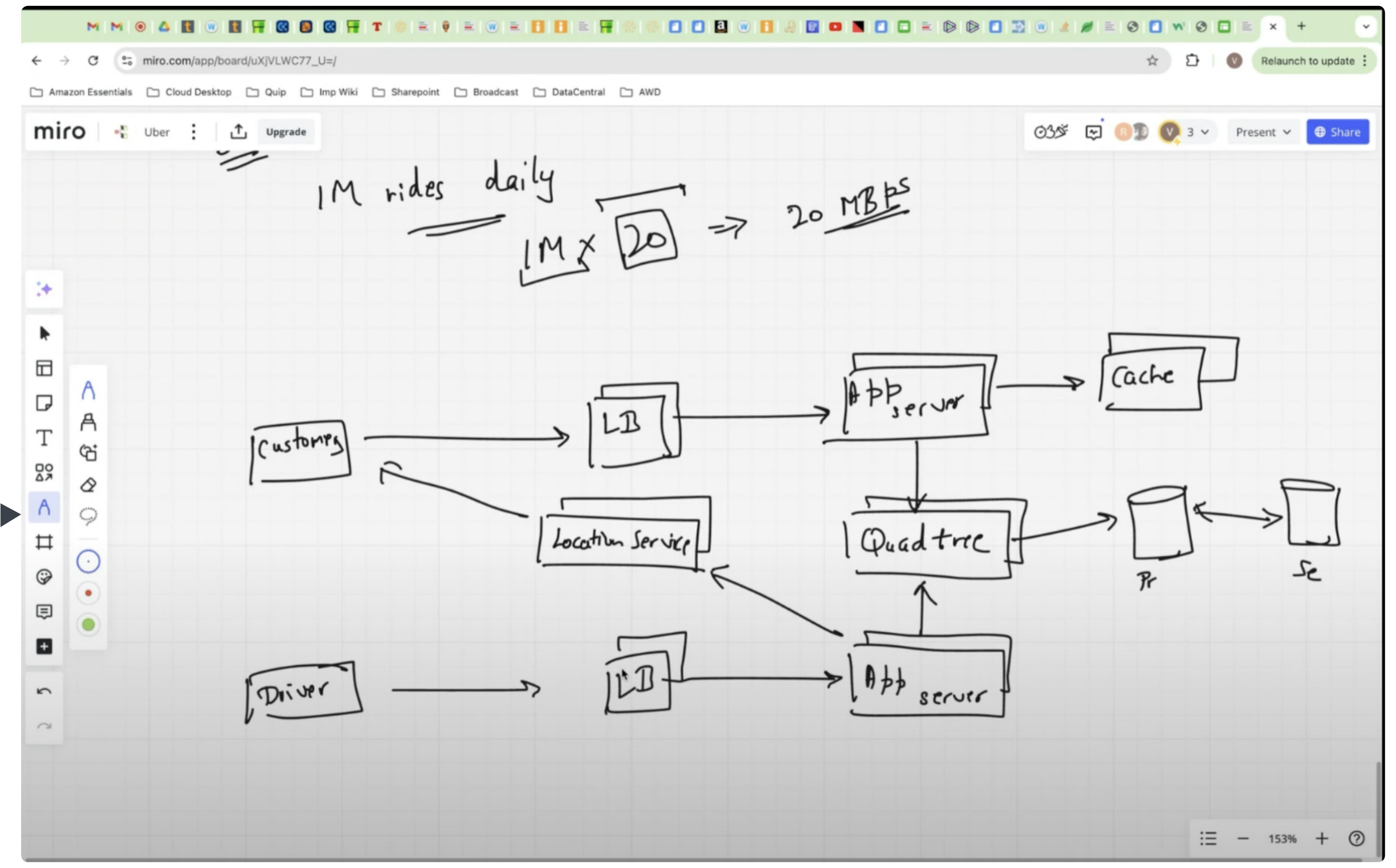
Drivers update Bandwidth(New lat,long   , DriverId) = 16+4=20 bytes
500k *20 bytes = 10 Mb/3sec

Driver after booking Design

1st way can be API call.

2nd way can be PubSub model . CustomerId subscribes to DriverId and driverId is publishing it's location in kafka

Partitioning can be done on DriverId or Ranking(for which driver to give preference)


miro.com/app/board/uXjVLWC77_U=/
miro
Uber
Upgrade
Present
Share
1M rides daily
1M × 20 => 20 MBps
Customers
LB
App server
Cache
Location Service
Quad tree
Pr
Se
Driver
LB
App server
153%