System Design - Ride Sharing System

Uber Eng.:
https://eng.uber.com/

SD:
https://www.youtube.com/watch?v=umWABit-wbk

Req. - Functional

Rider
~ See available riders in vicinity
~ Hail a ride
~ Notification of ETA/Location
~ Pay for a ride
~ Ride Pricing'
~ ETA Calc.

Driver
~ Notification of Book
~ Location of User
~ Driving Directions

Req. - Non-Functional (Write Heavy)

~ Real Time Data (Refreshed every 5s)
~ Location Accuracy
~ 10 M Drivers active daily
~ 100 M Riders active daily
~ 1 Ride daily
~ 99.99% Availability (1 hour/year outage)
(Backup Data Center)
Scale Calculations
~ Location Data 110M+
~ TPS 110M/5s = 22M TPS
~ 700 B Rides/day

Key Design Issues
~ Location Data Refresh/Accuracy
~ Scale of incoming location data
~ Consistency of location writes/reads
APIs
~ getDrivers(riderid, loc_lat, loc_long)
~ sendLoc(dirverId, loc_lat, loc_long, timestamp)

Datastore

Rider (NoSQL)
["riderId", "name", "location_lattitude", "location_longituge", "last_update_timestamp", "geo_cell"]

Driver (NoSQL)
["driverId", "location_lattitude", "location_longitude" "last_update_timestamp", "geo_cell"]

Rides (SQL)
["rideId", "riderId", "driverId", "rider_lat_long", "driver_lat_long", "dest_lat_long", "timestamp"]