

Functional Requirement

1. One to one chat **
2. Group Chat
3. Status seen **
4. Video call **
5. Read Reciep **
6. Profile
7. Notification **
8. Voice Notes
9. Picture
10. Contacts
11. Media Upload **

Non Functional Requirements

1. Reliable
2. Low Latency
3. Resilient
4. Consistent Order
5. Throughput
6. High Availability

Sequence Of Event

1. User will open Chat APP
2. Able to see all his messages
3. Able to see message of one to one chat and group chat
4. Able to chat and attach media with it

Capacity Estimation

1. Daily Active User is 2 Billion = 2×10^9

Daily Message transfer would be 100 billion

If one chat have 100 bytes and 10% is pics and 5% with video

1 pic size is 200KB and 1 video size is 5MB

10 billion pics and 5 billion video

$(10 \text{ billion} \times 200 \times 10^3) + (5 \text{ billion} \times 5 \times 10^6) + (100 \text{ billion} \times 100)$

$(10^{15}) + (25 \times 10^{15}) + (10 \times 10^{12})$

10PB + 25 PB + 10TB

35 PB per day message transfer

So for 5 Years = $5 \times 365 \times 35 \text{ PB}$

70 Exabytes for 5 years.

Bandwidth Estimation

For 1 day it is 35PB

1 million per day something is 11.57 / seconds

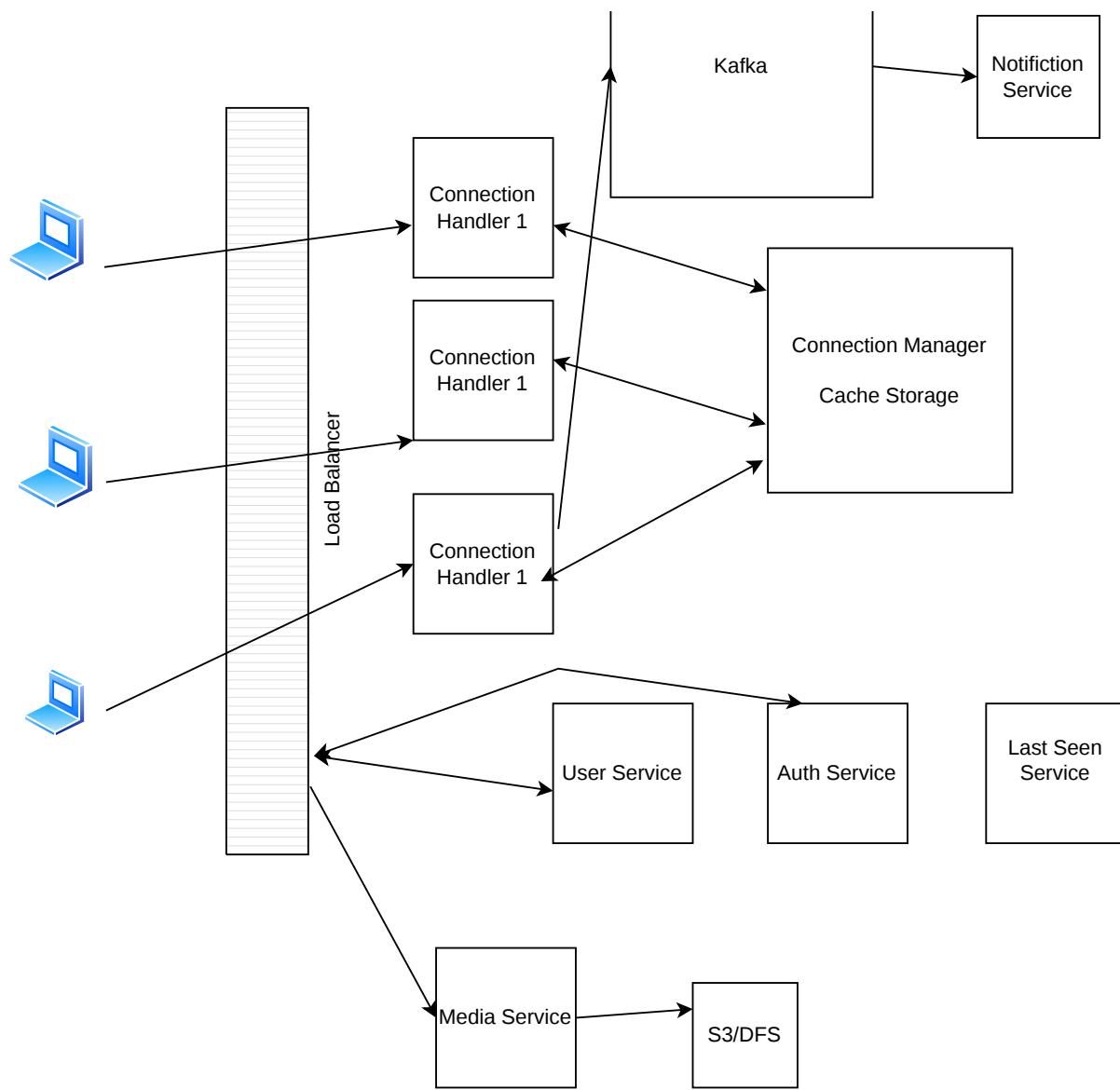
35×10^{15}

$35 \times 10^9 \times 10^6$

35 GB X 11.57

350 GB per seconds storage Ingress

Analytics Service



Databases
User Service - Will NoSQL document based db like mongo , couch db

Message Service - Write heavy db for this NoSQL columnar baseddb like cassandra

UBER HLD

Users

1. Real User who will book cab
2. Driver User

Functional Requirement

1. Book/Search for a cab **
2. Cab Types
3. See nearby cabs **
4. Set availability
5. Track Ride **
6. Share Cab
7. Estimating time of trip
8. Cancel Ride **
9. Notification *
10. Rating System
11. Payment Support **

Non Functional Requirements

1. Resilient System
2. Highly Scalable
3. In Payment there should be high consistent
4. In Cab booking or search there should be high availability
5. Low latency

Sequence of Event

1. User will open app
2. Able to search / book cab
3. Able to track driver location
4. Payment
5. Done travel . This is one is from real user side .

Capacity Estimation

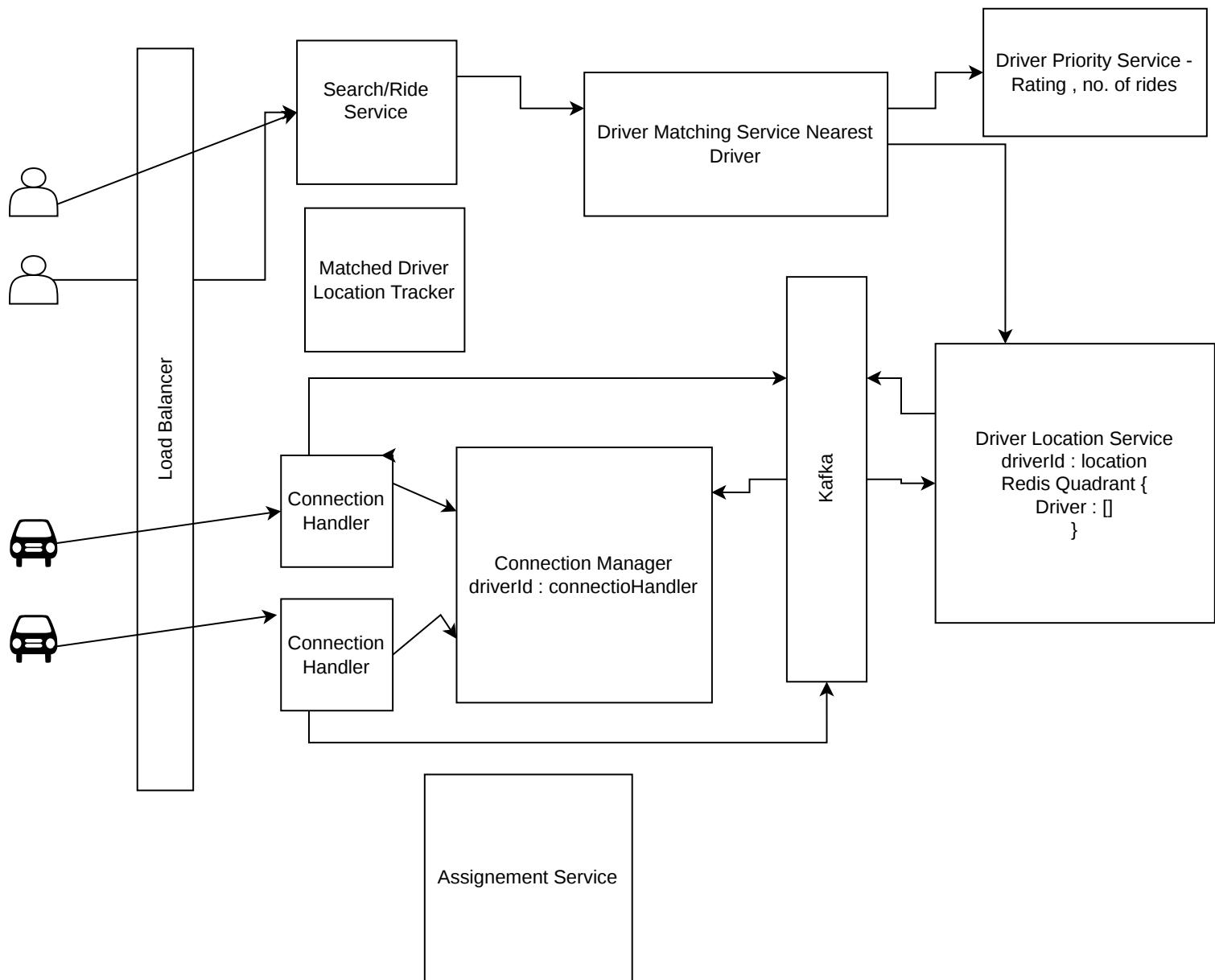
Storage Estimation

- Daily Active User 40 million per day
- All over world it has 10 million driver active daily
- 80 million rides per day
- 1 rides - destination detail , source detail , price , driverId .
- driver per unit location is also going to store like latitude and longitude .
- 1 unit of loc info let take 20 bytes every 3 second it is sending
- 10 minutes is sending - $10 \times 60 = 600$ second / 3 = 200 times .
- 200 times X 20 bytes = 4k bytes . 4 KB
- 50 bytes . 5KB for 1 rides data going to store

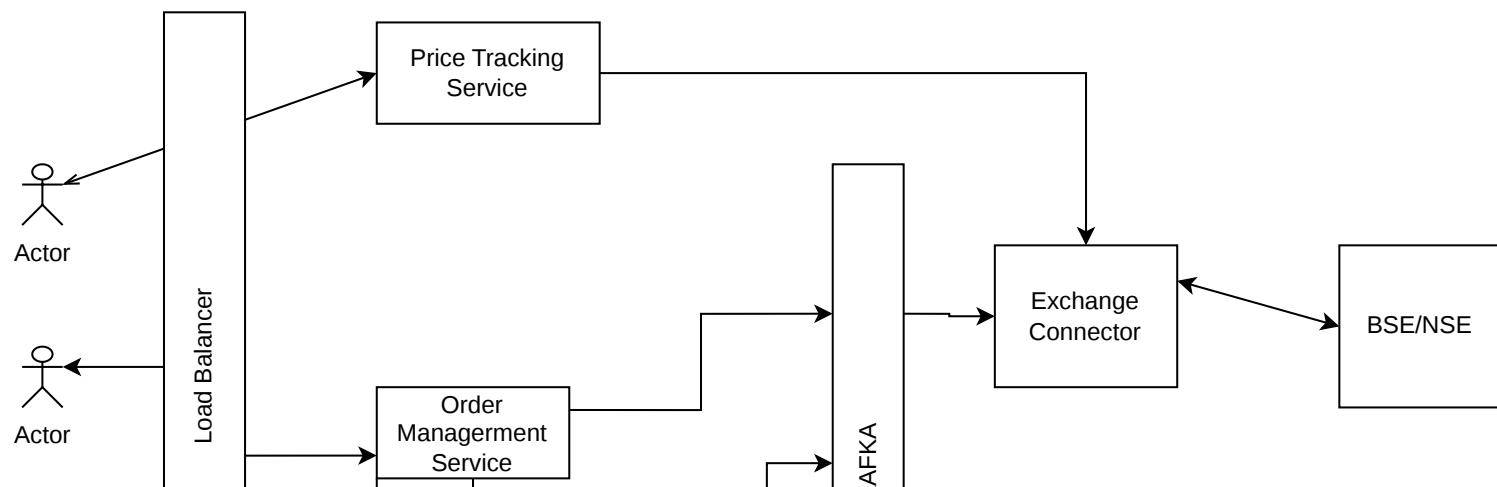
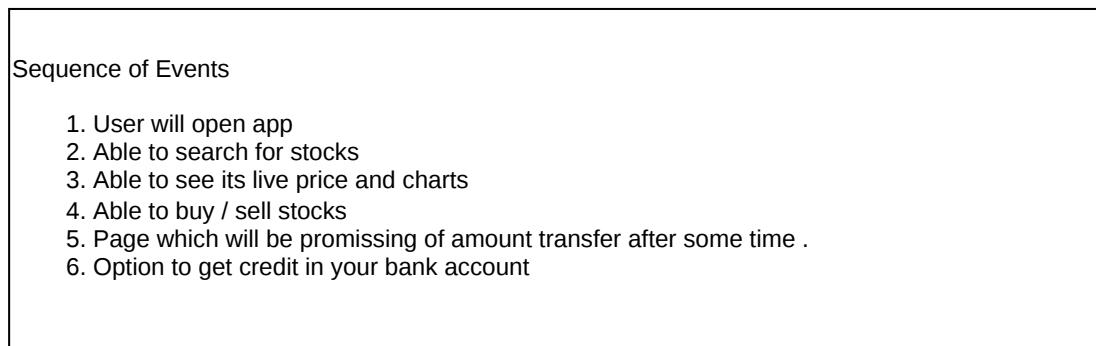
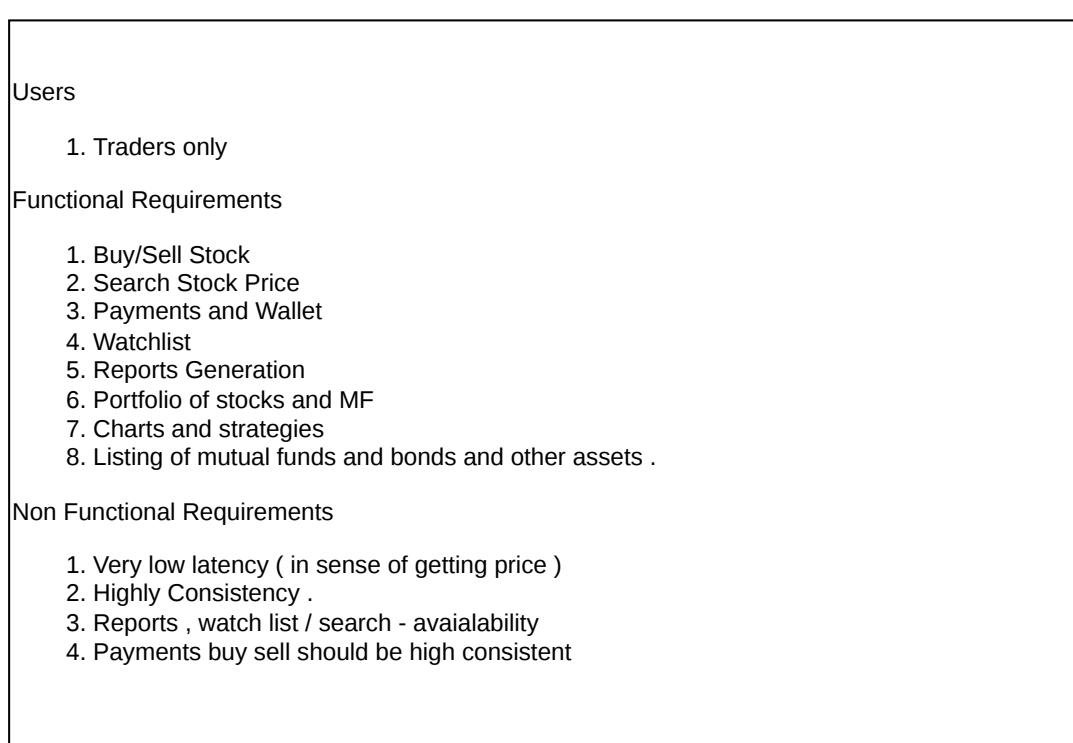
- 80 million rides per day
- $80 \times 10^6 \times 5 \times 10^3$
- 400 GB storage per day .
- For 5 years - $400 \times 5 \times 365$
- 8×10^{14}
- 800 TB

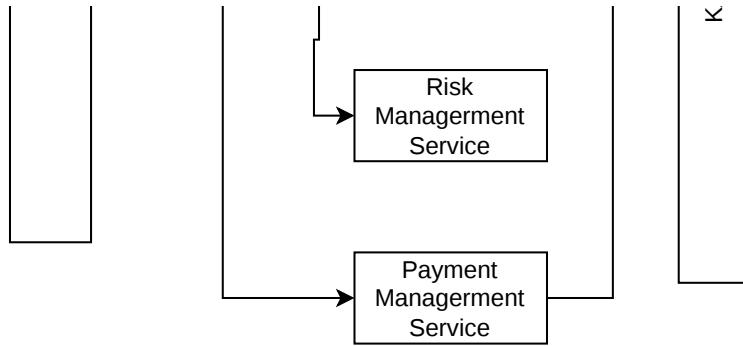
Bandwidth Estimation

- 400 GB per day
- 1 million per day = $11.57 \sim 12$ per seconds
- $400 \times 10^3 \times 10^6$
- $400 \times 10^3 \times 12$
- 4.8×10^6 bytes per seconds
- 4.8 MB/seconds storage ingress



Zerodha HLD





Orders

1. orderId , buy/sell price , Timestamps , quantity , status , bse/nse , order type - These details need to be very high consistency , but also ever increasing data so need horizontal scaling . So Would go with NoSQL document based DB like MongoDB
2. Watchlist - userId , stockId , watchListId - Schema are fixed , and data would also not very huge so would go with SQL database .

Capacity Estimation

Storage Estimation

- Zerodha has 16 million users on their platform
- 1 Million active user who trades daily
- 1 trades take 5 KB of data . stockid , orderId , price , bse/nse , user basic info , payment details .
- 1 user make 10 trades daily = $10 \times 5 \text{ KB} = 50 \text{ KB}$
- For 1 million user = $1 \times 10^6 \times 50 \times 10^3$
- 50 GB storage per day going to store .
- For 5 Years = $50 \times 5 \times 365 \text{ GB}$
- 100×10^{12}
- 100 TB of data for 5 years .
- There also be user data stored for 16 million user . kyc detail , account detail , personal info . So 5 KB of data . for 16 million
- $16 \times 5 \times 10^9 = 80 \text{ GB extra data .}$

Bandwidth Estimation

- 50 gb per day .
- 1 million per day will give $11.57 \sim 12 \text{ per second .}$
- $50 \text{ GB per day} = 50 \times 10^3 \times 10^6$
- $50 \times 10^3 \times 12$
- 600×10^3
- $0.6 \text{ MB / second .}$
- 6 hours run time only .
- Need to multiply with 4 . Ao 2.4 MB/second

Youtube HLD

Actors

1. User who will just watch videos
2. Upload who will upload and watch videos

Functional Requirements

1. Stream and Watch Video **
2. Upload a Video **
3. Search for a Video **
4. Recommendation **
5. Like Share Comments and Subscribe Video **
6. Super Chat
7. Suppor Devices - Android , Web , iOS , TV
8. Playlist
9. Playback Speed
10. Video Quality **
11. Captions
12. Save the last seen video
13. Subscribe and Notification
14. Live Streaming
15. Trending
16. Forwarding Video
17. User Profile **
18. Advertisement
19. Encoding/Decoding Video

Non Functional Requirements

1. Highly Available .
2. Low Latency
3. Resilient System
4. Less Buffering

Sequence of Event

1. User will open App
2. He should able to see few videos as per recommendation
3. Able to search for some video
4. Able to stream it
5. Able to upload video .

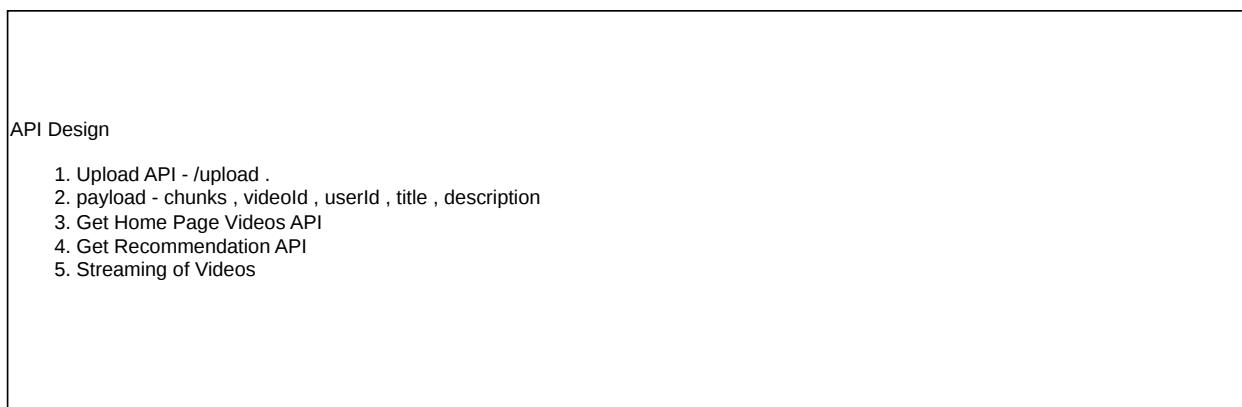
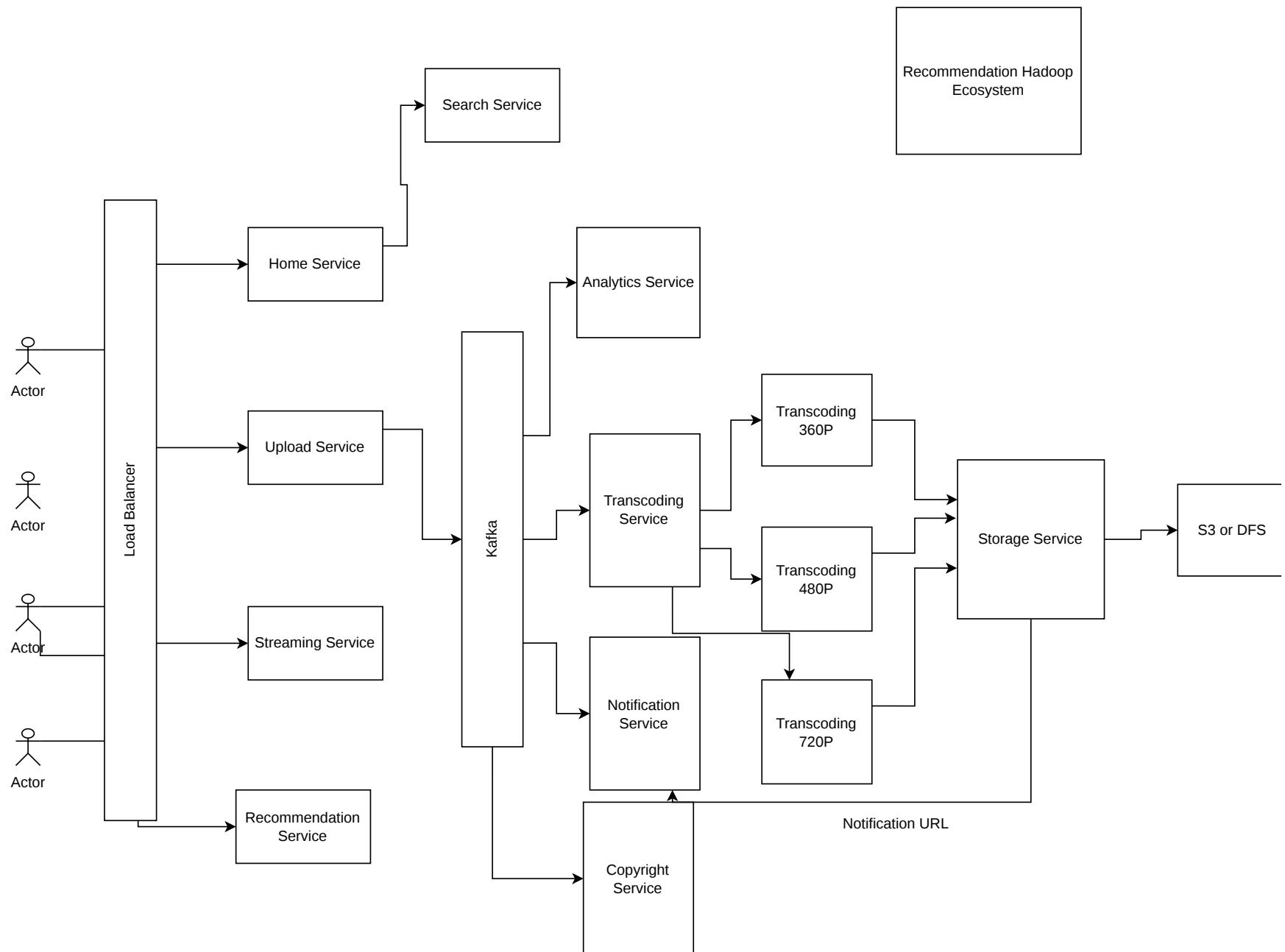
Capacity Estimation

Storage Estimation

- DAU Daily active user is 2 Billion .
- Ratio of user uploading video and just watching is 10:90 .
- 200 million uploaders are there .
- Let's take 1 video size is of 100MB and there would be transcoding also there
- Per user 500 MB of storage .
- There will be metadata also like userId , timestamp , transcode option available , description , title , tag . metadata size is 10KB .
- 200 Million X 500 MB per day
- $200 \times 10^6 \times 500 \times 10^6$ Bytes
- 100×10^{15} bytes .
- 100 PB per day
- For 5 Years = $5 \times 365 \times 100$ PB
- 200 EB for 5 years .

Bandwidth Estimation

- 100 PB per day .
- 1 Million bytes per day will give 11.57 bytes per seconds . ~ 12 bytes per second
- 100×10^{15}
- $100 \times 10^9 \times 10^6$
- $100 \times 10^9 \times 12$
- 1.2×10^{12}
- 1.2 GB per seconds going to store data .



]