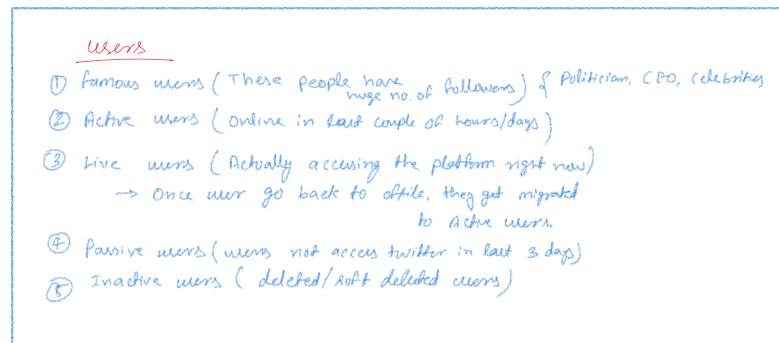
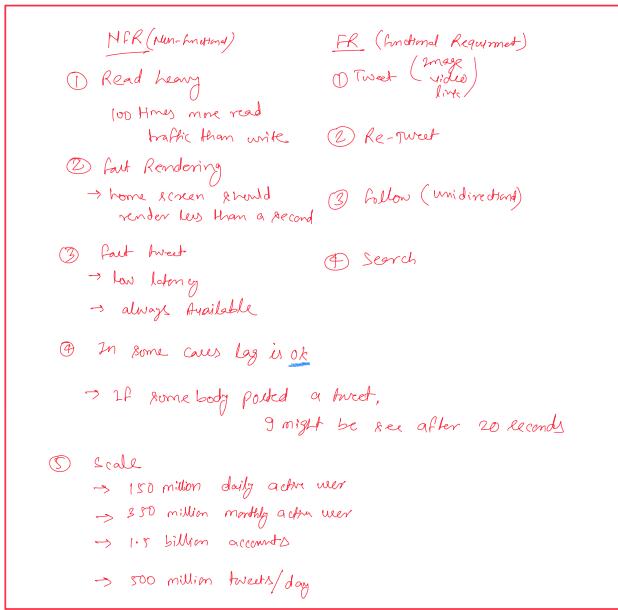
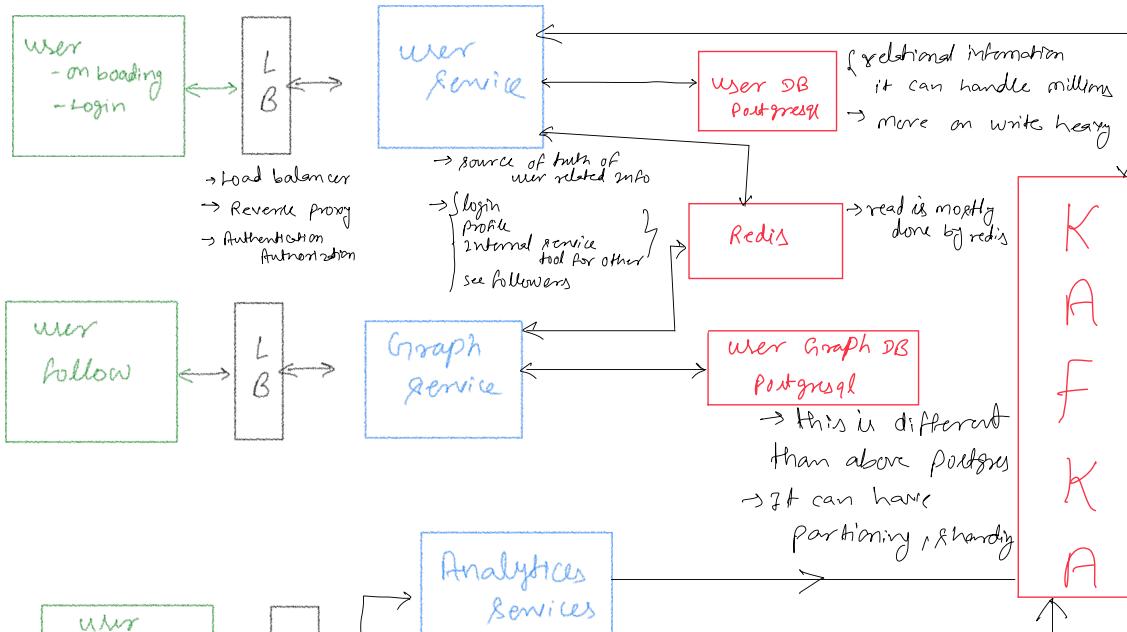
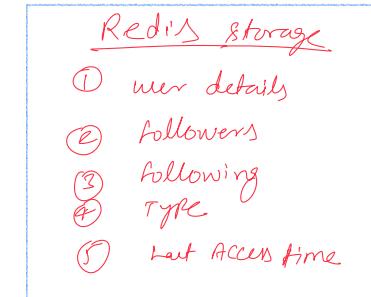
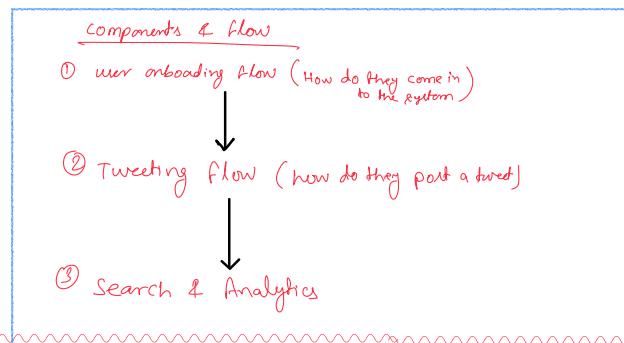


# Twitter system design



Points to remember

- ① If system is Read heavy, it's better to keep data in cache, or precompute the information.
- ② It is not good to fetch tweets of user whom he follows. coz query on database on such scale is not feasible.
- ③ So, we need to cache the users timeline.



User service

- whenever someone tries to look up user info, it will first look into Redis.
- if Redis doesn't have, then it queries to PostgreSQL and keeps cache in Redis.

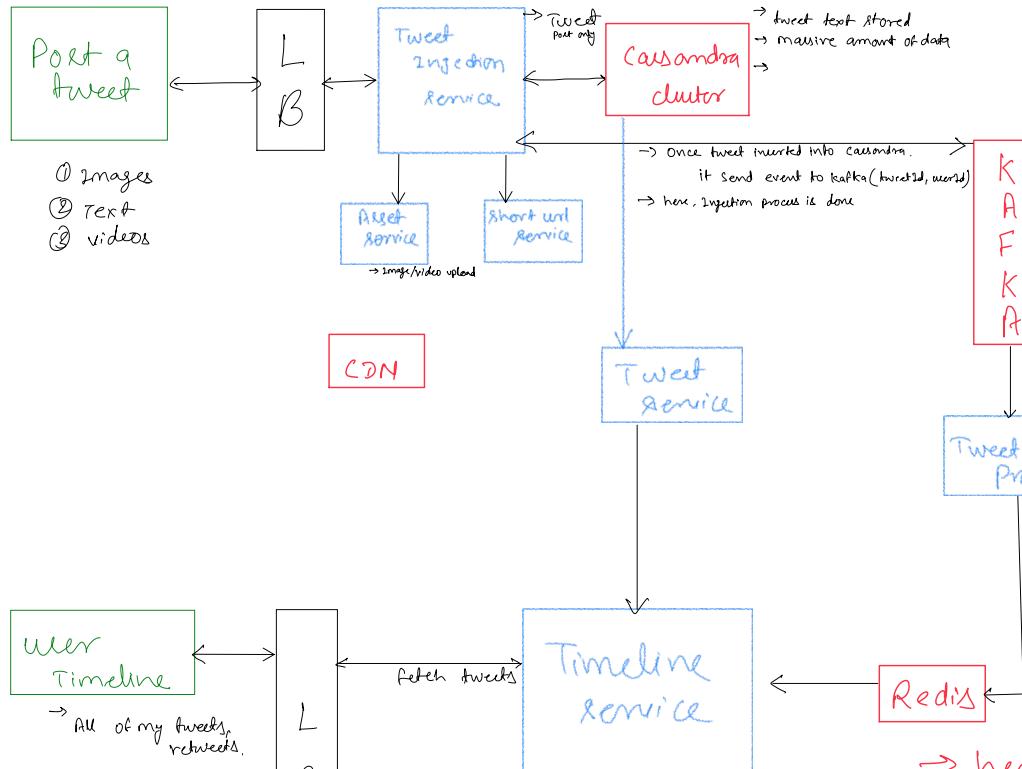
Graph service

- To fetch all the followers of a person
- The reason being called it as "graph", because everybody are connected to each in this ecosystem.



- here are the user who are live now.
- if someone posted something and this user following him, then it's better to show them this post.
- if somebody tagged me in a post
- if any event occurred, we need to share with this user via websocket.
- if user is not live anymore, then send this information to kafka, and then send back to redis, and mark that user as active
- Live state → Active state
- and other system can use this information to change the behaviour of services

- here, the database is postgres, but heavily sharded.
- { user\_id, follower\_id, created\_at }
- This database does not update frequently
- here we can also use redis to store followers, following.
- Again, it will first query redis, if don't find then query database.



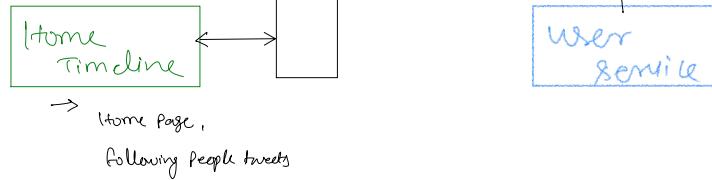
### Tweet Ingestion Service

- it only stores tweets, do not have GET APIs
- Once, tweet goes into cassandra, it releases an event to kafka.

### Tweet Service

- it is a source of truth, and sits on cassandra
- This service will provide all the sets of APIs to fetch tweets.
- This service also generates timeline of an user.
- Query the graph service to find the list of user to create their timeline

→ here we will face a problem

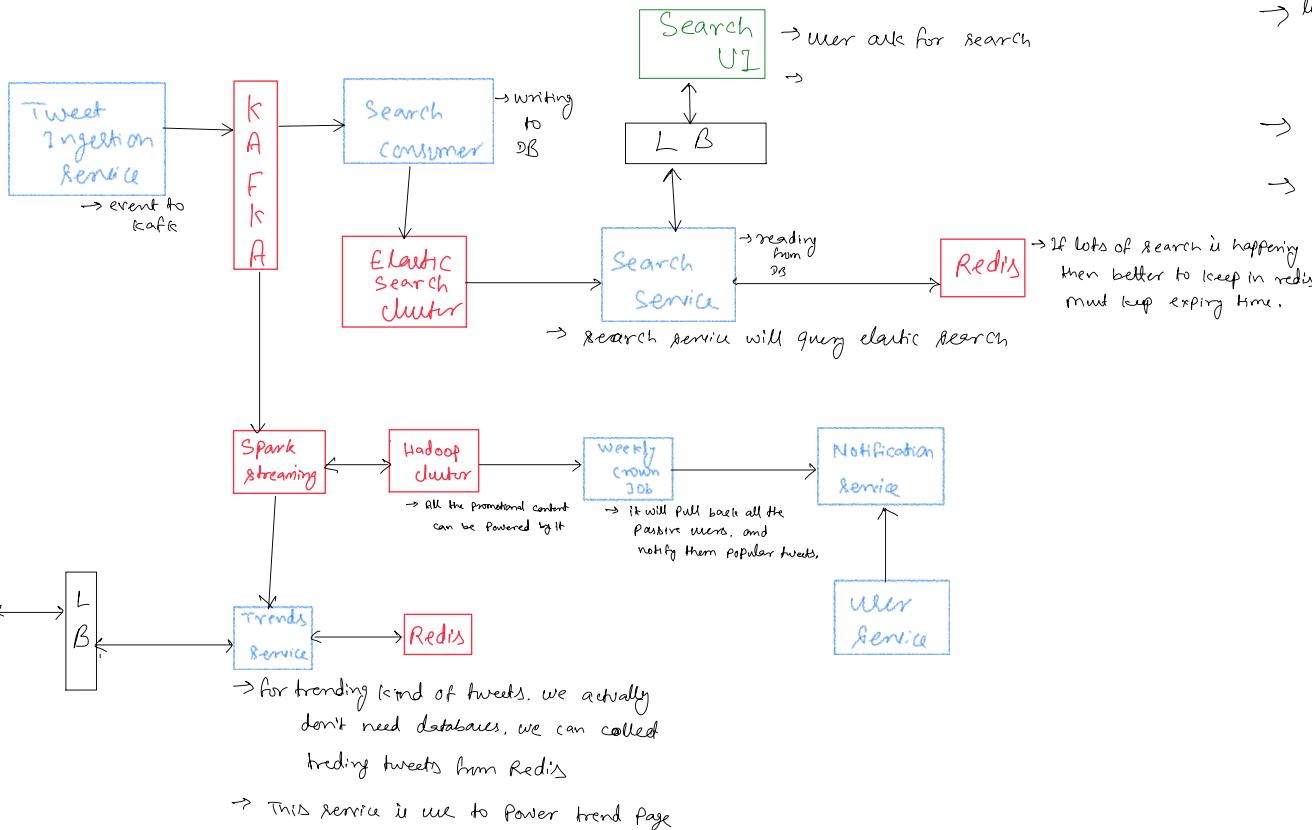


- we will have a problem if we keep all user timeline, then cannot handle that much of pressure.
- it won't be efficient
- We will store only their timelines, which are active.

how can we handle, if Donald Trump Tweets

- The above solution won't work.
  - Timeline Service knows that redis have tweets of normal people  
it calls Graph Service to fetch famous person following
  - And get their tweets and merge the user's timeline tweets

# Search and Analytics



## Consumer Service

- looks into all the tweets coming into the system, and stores in elasticsearch database, which will use to search tweets
  - Consumer Service write data to dB
  - Search Service reads data from dB.





