

Instagram Newsfeed System Design

A Functional Requirements

- ① Newsfeed contains photos/videos from people you're following
- ② Reflect suggested Posts - P1 requirement
- ③ Newsfeed should get updated regularly
- ④ Notification for new post - P1 requirement

B NFR

- ① System is always available
- ② Generate newsfeed in realtime, Once user opens the App. updated newsfeed should be reflected.
- ③ New post from people should be sent to followers in min latency - 10 seconds threshold.

C Capacity Estimation

① Traffic Estimates

Insta - around 1.3B active users

assume - 50% login daily = 650 million

on average, access newsfeed 10 times/day

Traffic / Day = 6500 million Requests / Day

TPS = 75 K requests / sec

② Storage Requirements

avg size of photo = 10 KB

video = 2 MB

Precompute 200 items to be displayed in feed

175 photos
25 videos

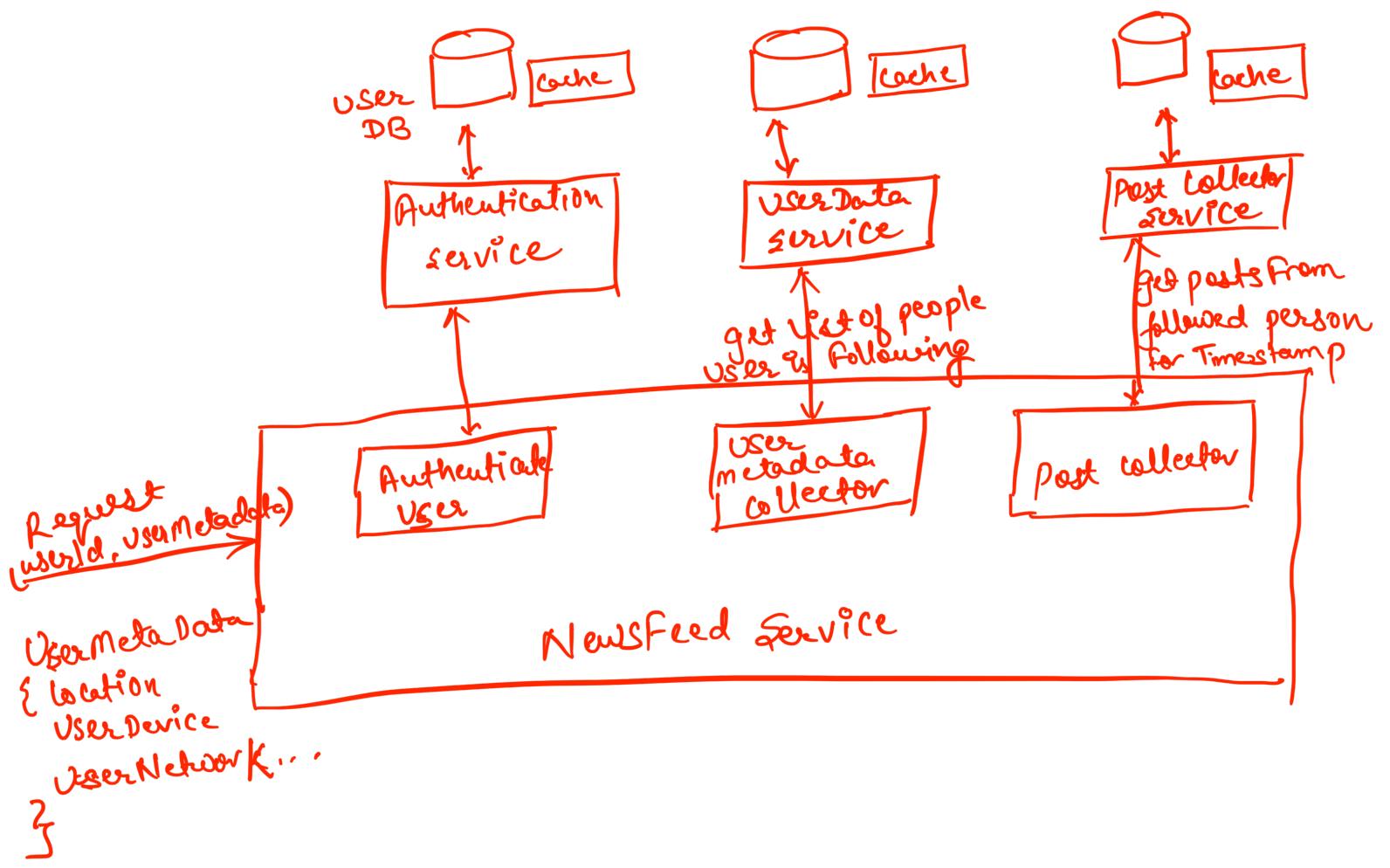
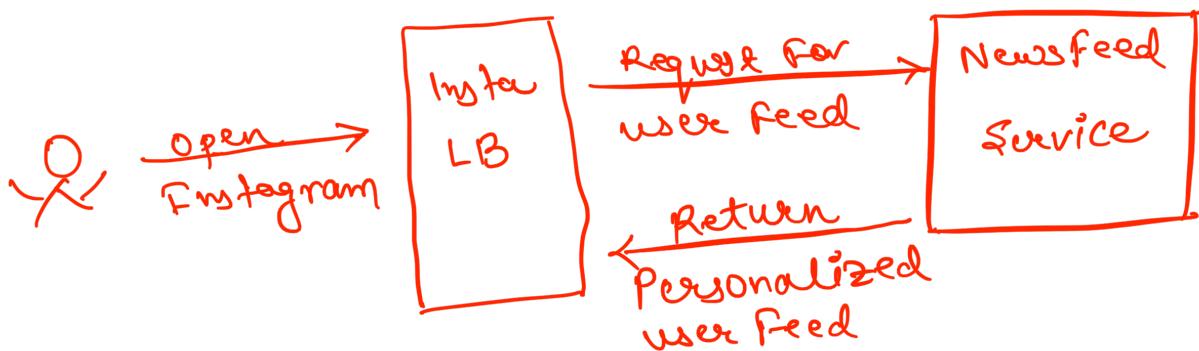
Total storage for single user =

$$175 * 100 + 25 * 2 * 1024 = 0.07 \text{ GB}$$

For all users = 650 million * 0.07 GB

$$= 45,500 \text{ TB} = \underline{46 \text{ PB}}$$

D High Level System Design



Possible Bottlenecks :

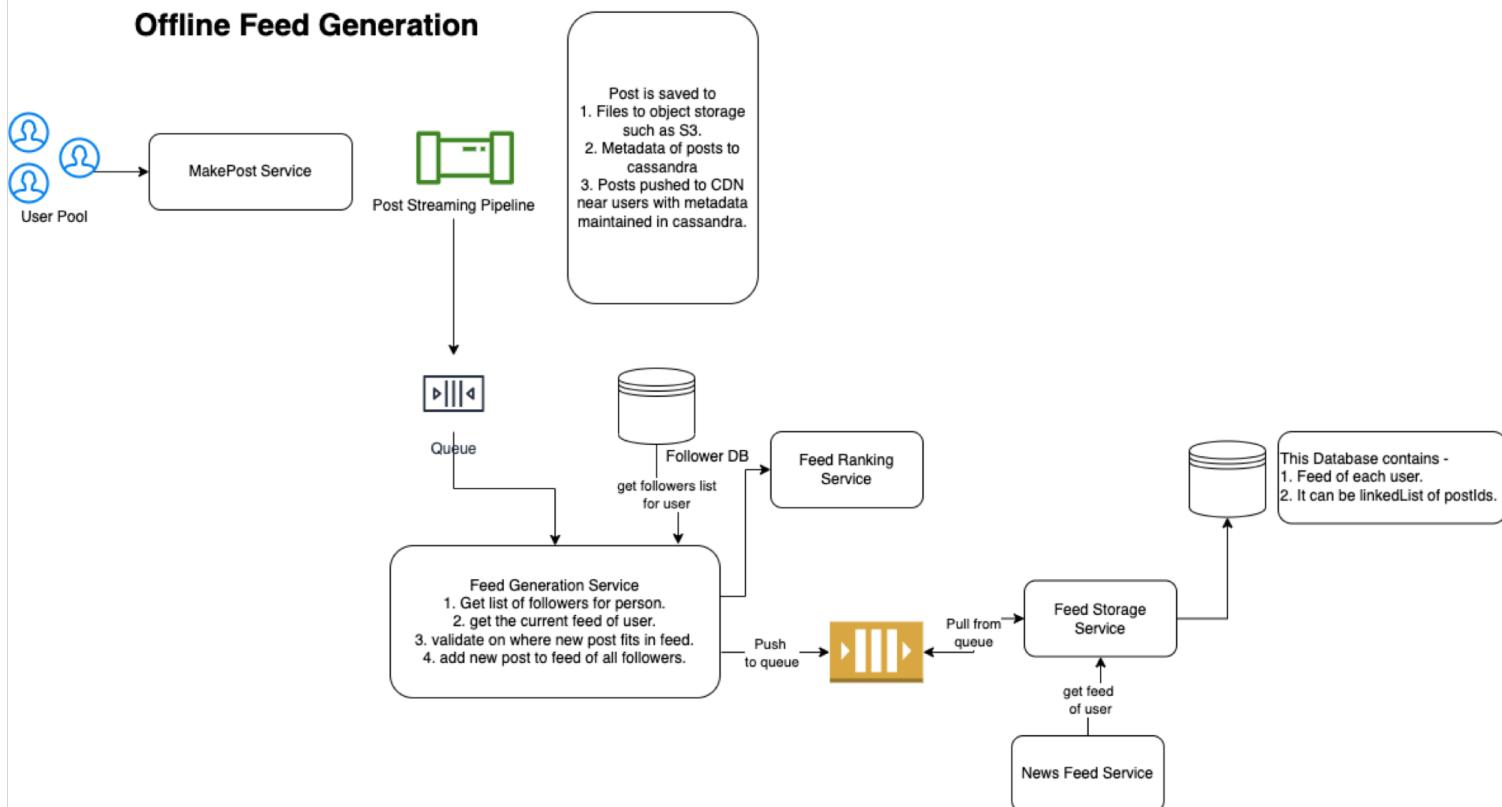
→ All the computations are done in realtime, which could be time consuming for the scale of Instagram.

<THINK> Can we pre-compute the Newsfeed ???

The feed is created from the posts of users
Can we create/update userfeed when someone makes a post and serve the feed on request?

OFF LINE FEED Generation

Offline Feed Generation



Feed Publishing to Users

- ① Pull → As in diagram, user requests for new feed, and gets update feed in response.
 - Need to define interval on which Pull request should be made.
 - Optimal interval derivation could not be possible
 - Either extensive resource utilization at server
 - OR no post update in real time.
- ② As a little optimisation, use long Polling.
- ③ Explore WebSockets | Server Sent Events (SSE)

For scale required at Instagram, we continuously update the feed, so keeping a persistent connection open for online user makes sense.

→ New feed items keep on appending based on user interests.

what about users with lot of followers?

Publishing new post from celebrity user could be much more resource intensive. So instead for such users, we can rely on pull based approach.

This way, system would work in Hybrid fashion depending on Type of user.