# Amazon Prime Video

## functional

1. upload video
2. search video
3. view video

## Non functional

1. uploads should be fast
   ↳ (multi hour video & if customer have to wait long hours it won't be good)
2. view video
   (no or minimal buffer)
3. Low Latency
4. high availability → eventual consistency
   ↳ because it is ok if a video is seen by few users after sometime but once available video should not go anywhere.
5. Reliability ⇒ video should not be lost from ecosystem

additional features → showing view count
→ adding wish list
— etc....

$\boxed{\text{CAP Theorem}}$

100 millions monthly active users → assumptions

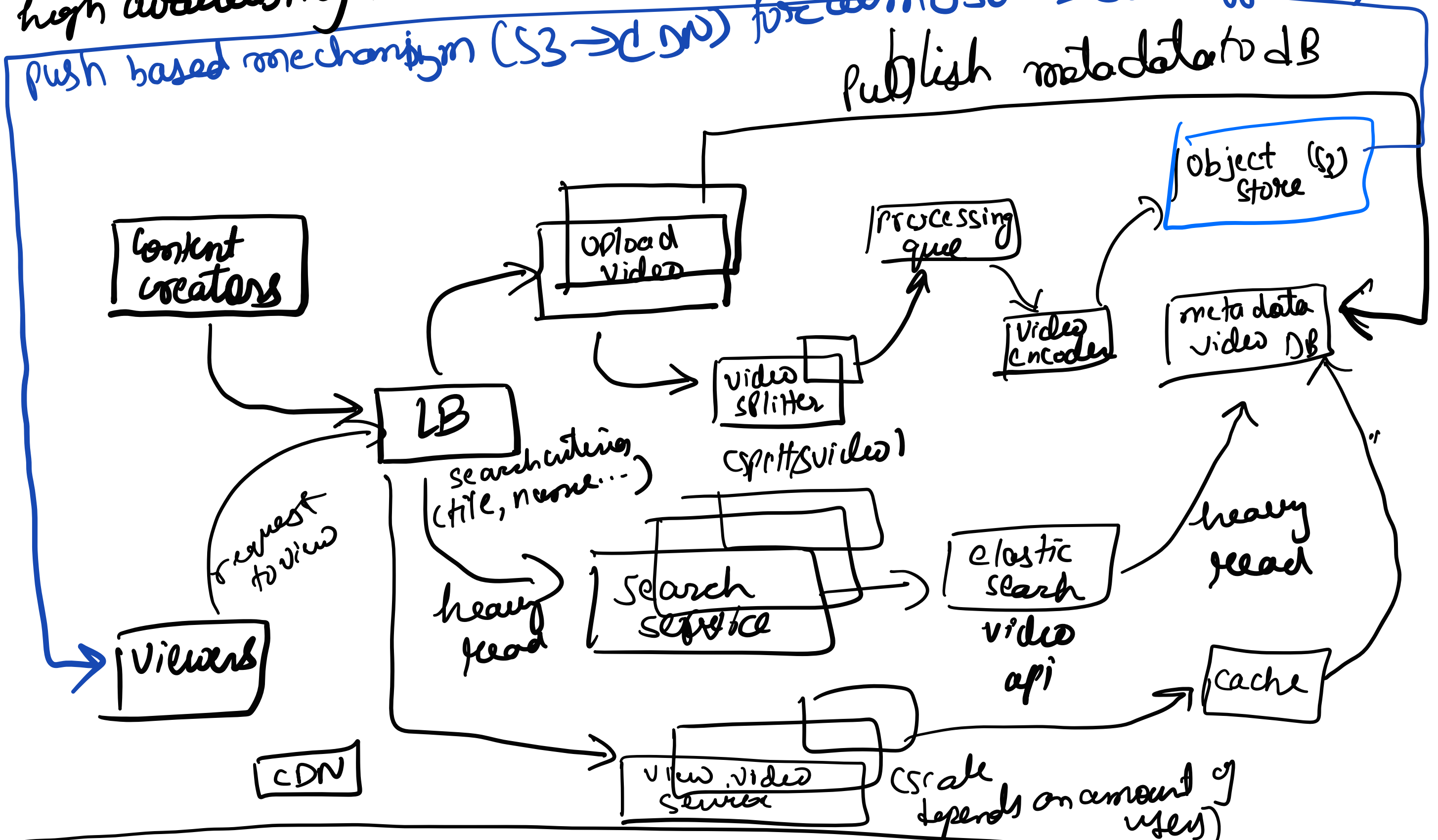
↓

1:100 upload to view

pull based cache mechanism in search as search can take

we can do partitioning of video depending on need

high availability, eventual consistency (Push during non peak hrs) → to avoid throttle (but delay in availability)



100 million 1 million

100:1 → 1% to 100% upload users (1 million at time)

(LB) load balancer needed → Assumption of 100 million users will need more clusters of video service

upload video service → upload video from end users

S3 → object storage to store video

metadata DB → to store video metadata

how to make upload fast

we can split video let say 10 minute long each
& distribute it using map-reduce concept

video encoder → ① Perform aggregation of splitted video
② store video in multiple formats (mp4, ...)
③ store for across multiple devices
⋮

Q what if amazon prime launched high trending video?

~~DO~~ Will not change the design much
we can add manually / force push of certain
contents (depends on push frequency)
this can help in attain this

or put cache TTL to very low for trending
days to make video availability

## Notes

Start with problem statement
↓
write down functional & non functional requirements & key tradeoffs
↓
Implement functional requirement
try completing one & then move to

Talk about
→ non-functional
requirement

requiren

→ ask for feedbacks or questions

→ co-relate your services to show reusability

→ Use SOLID/OOPS while designs

another so that if time runs ou

I still have competed some

↓

Jump to another functional requiremt

↓

⋮

↓

$\boxed{\text{done}}$