Video Streaming Service

(Youtube,

Netflix,

Amazon Prime,

HotStar

Ullu 😊

)

Functional Requirements:

1) Uploading a video
2) Playing a video
3) User's Home Page
4) Search
5) Support on all devices

Non-Functional Requirements:

1) NO Buffering (Low latency + High Availability)
2) User's Session Time (" ")

Scale:

DAU: 1B / day

uploads per min : 150k

views per min : 500M
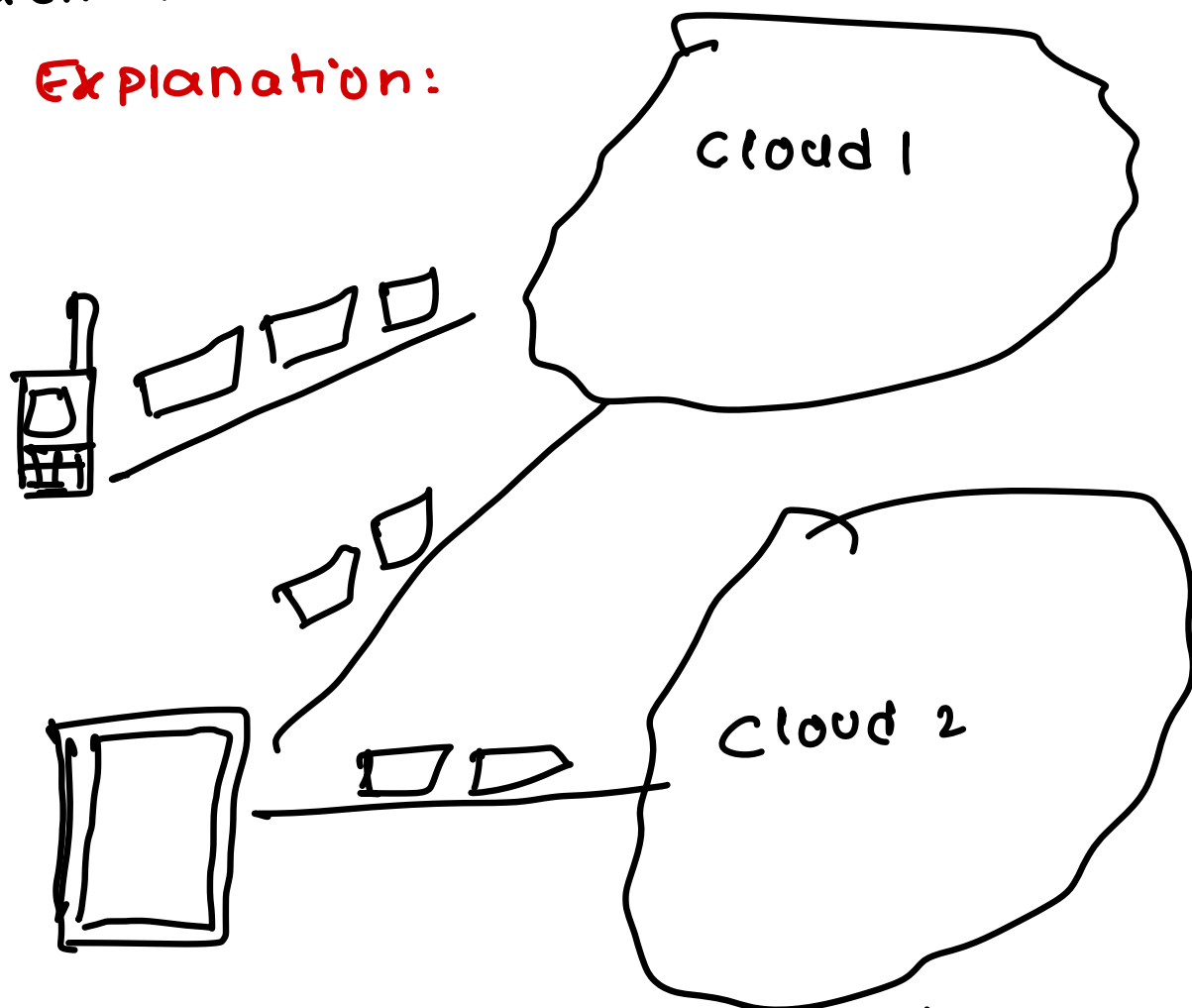
MAU: 3B

1) Supporting Multiple Devices:

Devices	Format (i)	Dimensions (j)	Bandwidth (k)
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:

$\Rightarrow$ we get $i * j * k$ combinations to support

Types of Users:

1) Clients (Mobile, Laptop, IPad, Tablet etc...)

2) User (Men, Women, gay)

3) Production Houses

Chunks Explanation:

Cloud 1

Cloud 2
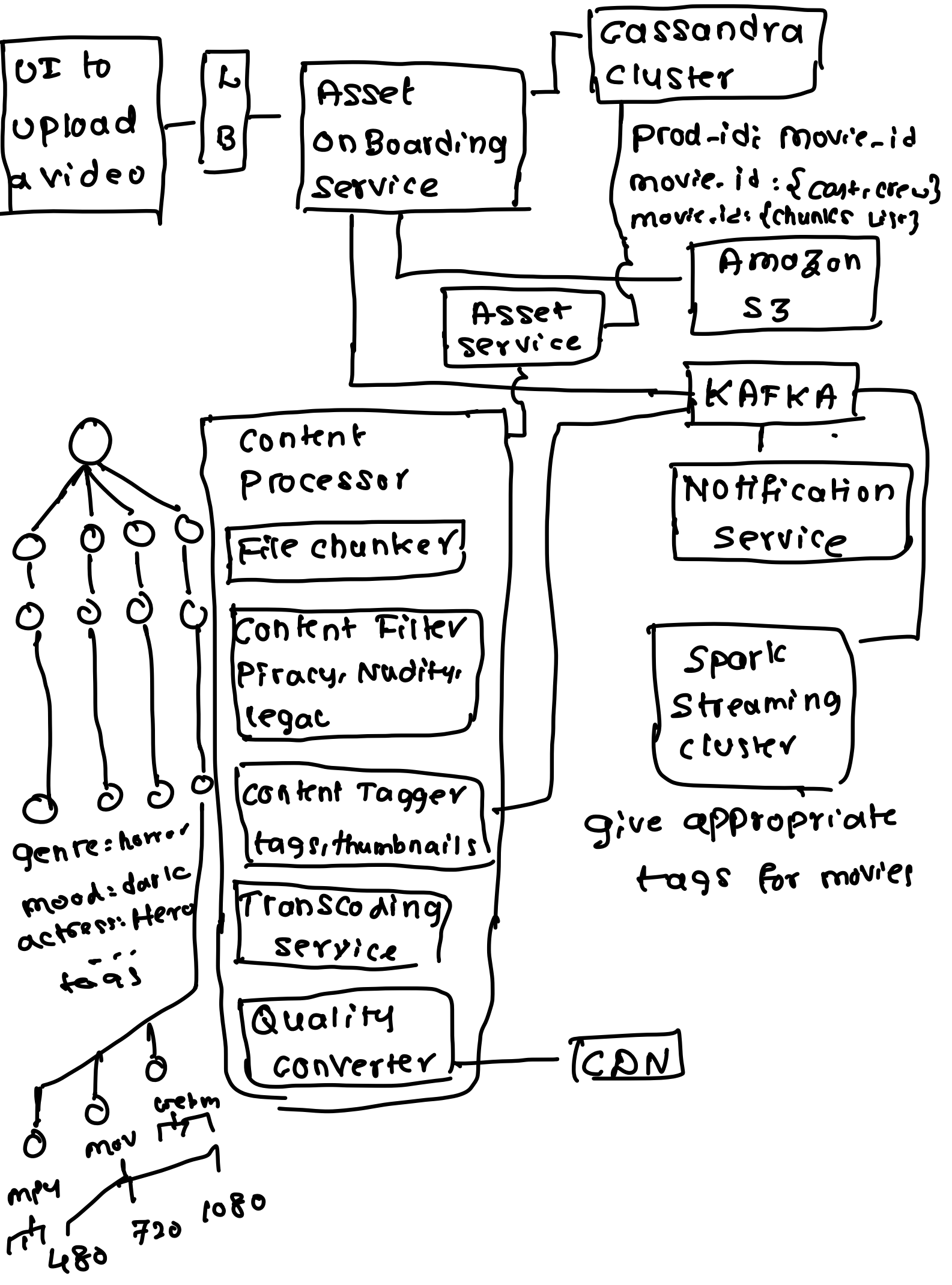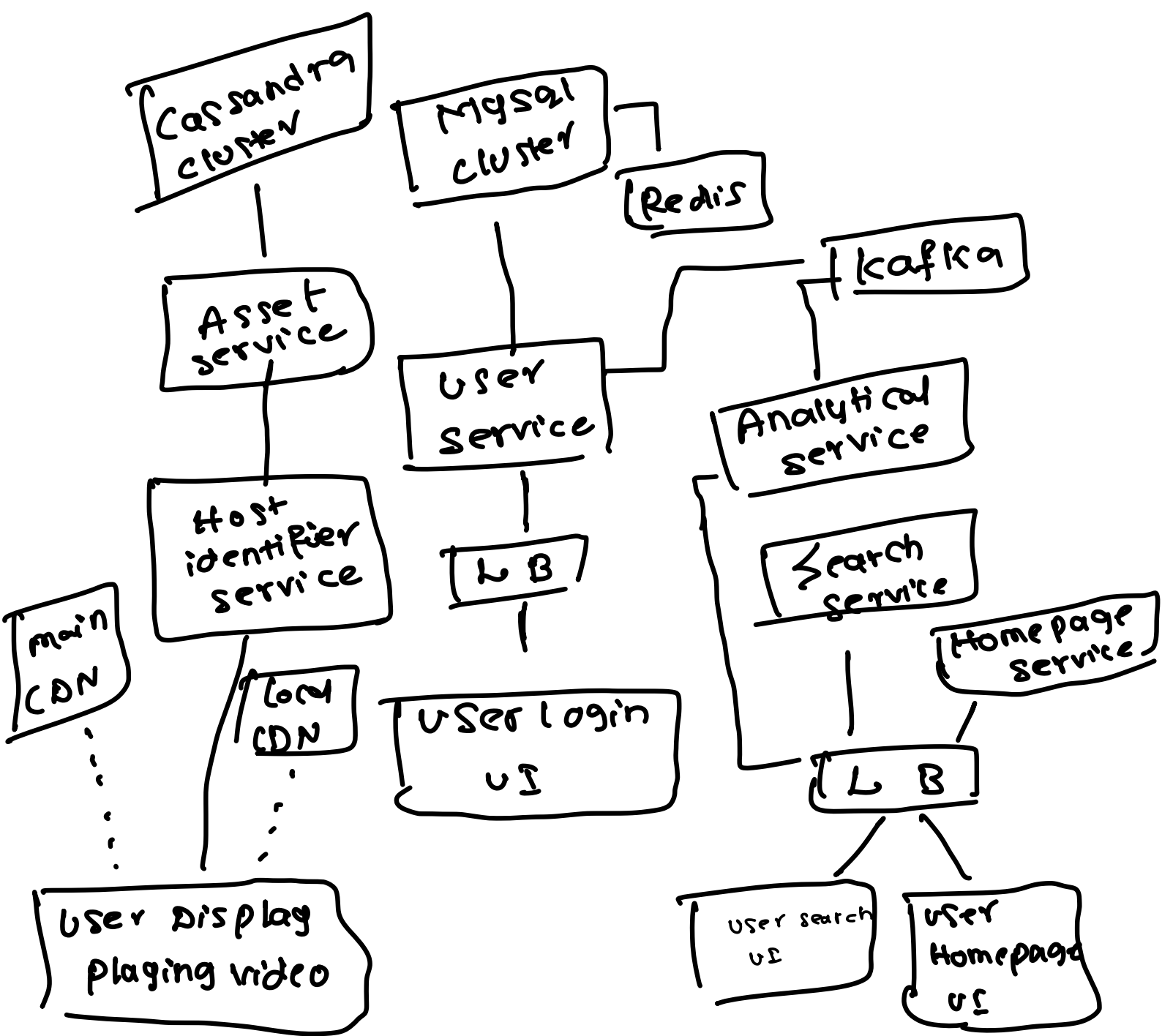
Clients process chunks of data but not the full video. The processing depends on bandwidth, format the device supports along with the dimensions

Bandwidth can be flluctuated at times. Therefore, clients process the video based on the bandwidth

Production Houses uploading a video (System Arch)

UI to upload a video — **LB** — **Asset OnBoarding Service** — **Cassandra Cluster**

Cassandra Cluster:
- Prod-id: movie_id
- movie_id: {cast, crew}
- movie_id: {chunks urls}

Amazon S3

Asset Service

KAFKA

Notification Service

Spark Streaming Cluster

give appropriate tags for movies

Content Processor
- File chunker
- Content Filter — Piracy, Nudity, legal
- Content Tagger — tags, thumbnails
- Transcoding service
- Quality Converter

CDN

genre: horror
mood: dark
actress: Hero
tags

mp4   mov   webm
480   720   1080

User Login, Search, Homepage, Play Video:

Cassandra cluster

Mysql cluster

Redis

kafka

Asset service

User service

Analytical service

Host identifier service

LB

Search service

Homepage service

main CDN

local CDN

User login UI

LB

User Display playing video

User search UI

User Homepage UI

User Search Service, Homepage Service, Analytical Service:

```
KAFKA ── Spark Streaming Cluster ──← Hadoop Cluster ── Spark Jobs/ ML
```

Identify appropriate tags for a movie

```
Search Consumer
    │
Elastic Search Cluster
    │
   [LB]
    │
Search UI
```

User service

```
Spark Jobs/ ML
    │
  KAFKA
    │
Recommendation Service
    │
Cassandra Cluster

userid: {movies watched, possible watchlist}
```

User Profiling

Traffic Predictor

```
Homepage Service
    │
   [LB]
    │
Homepage UI
```

```
KAFKA
    │
CDN writer ── · · ·
    │         │
 [LCDN1]   [LCDN2]
```
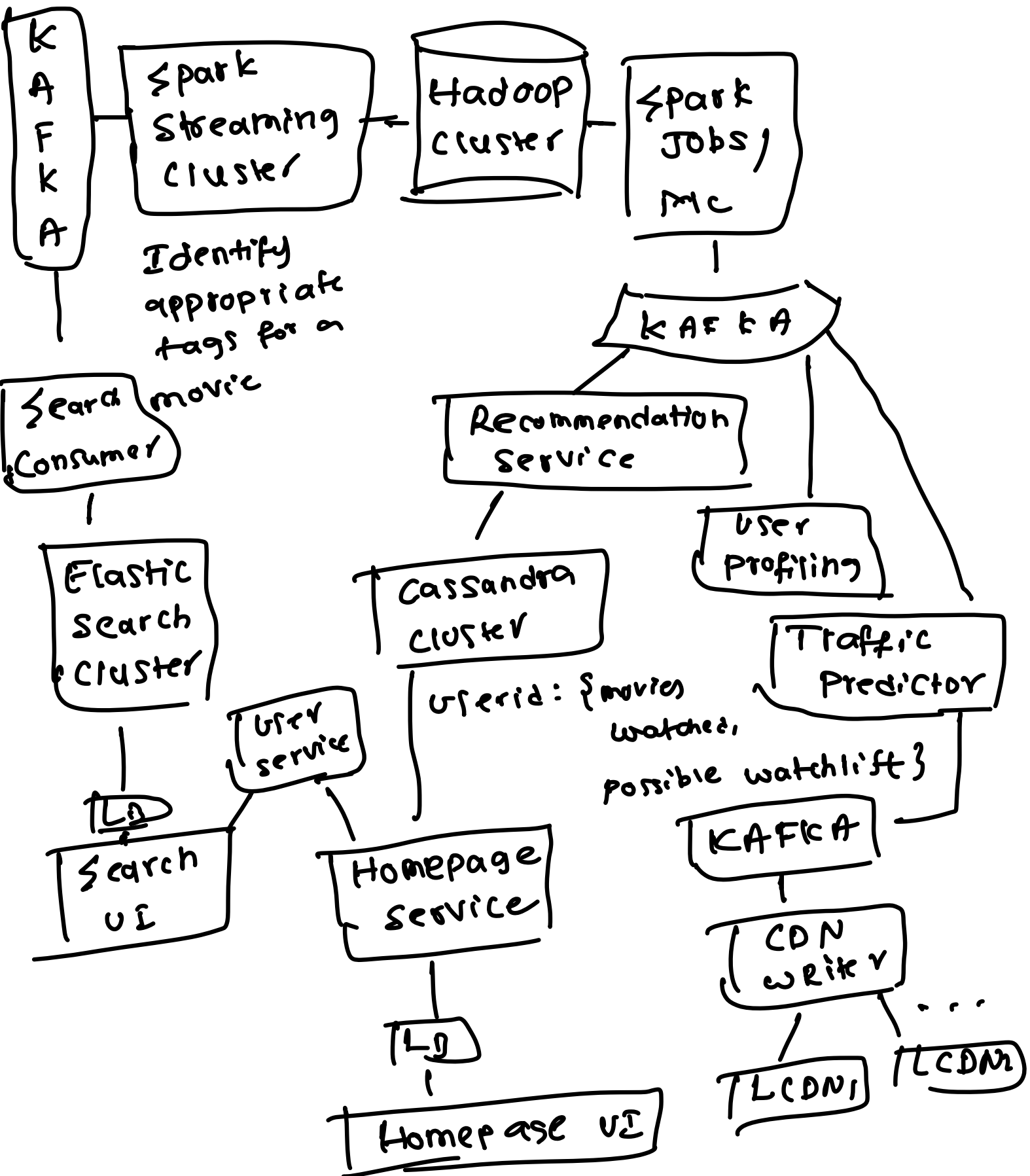
Decentralization via CDN writer:

Replica1 S3   S3   Replica 2 S3

LCDN1   LCDN2 ... ~1 LCDN n

SPOF & Bottleneck
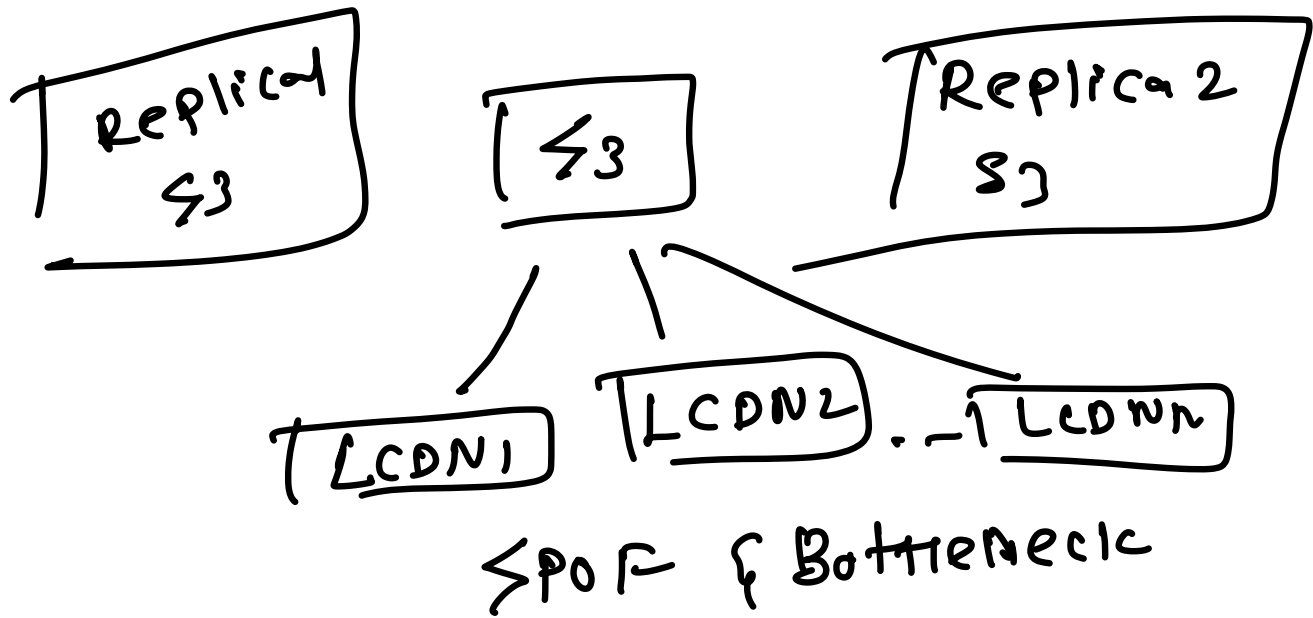
∴ Upload chunks across CDN1

LCDN1	LCDN2	LCDN3	LCDN4	LCDN5
C_{1-25} C_{1-5}	C_{5-10}	C_{10-15}	C_{15-20}	C_{20-25}

UI — Modem — Router — ISD ...... o—o—o CDN / CDN

U_2
U_3
...
U_N

Cache maintained by Netflix